DAVIX: A TOOLSET FOR DATA ANALYSIS AND VISUALIZATION

A

PROJECT

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER of SCIENCE

By

Amanda L. Gellhouse, B.S.

Fairbanks, Alaska

April 2014

Acknowledgments

Academia never operates in a vacuum and it is with pleasure that I have had the opportunity to be inspired by a number of individuals through the course of this project and my career.

I would like to thank foremost my undergraduate computer science professors at the University of Alaska, Fairbanks, who fostered my interest in the field and made me pursue it wholeheartedly; these include Dr. Kara Nance, Dr. Chris Hartman, Dr. Mitchell Roth, and Dr. Peter Knoke.

I would like to thank my employer, Wostmann & Associates, Inc., for their support throughout this process, including offering flexibility in my work schedule and encouragement of my studies.

I would like to thank my family and friends for their patience and support, in particular my brother, Nils Gellhouse, for his ceaseless championship of my efforts.

My graduate committee has been indispensible; Dr. Kara Nance has provided expert guidance, cultivation, and feedback at all stages of the project, and Dr. Brian Hay has shared his expertise in virtualization and provided assistance at key stages of the project. I would also like to thank Dr. Jon Genetti for his support and guidance throughout this project.

Finally, this project would be all for naught if Raffael Marty had not expressed interest in upgrading the original DAVIX; his enthusiasm and expertise has kept this project moving forward. It would also not have been possible without the significant contributions of Edward McCabe who volunteered as a developer, Justin Searle, who made himself available for knowledge-transfer when we were examining similar toolsets, Nancy McClure, for assisting with our logo design, and Matt Jones for volunteering resources and seeing grander potential in the toolset than we had even begun aspiring towards.

Table of Contents

Section 1 Introduction

DAVIX (Data Analysis and Visualization Linux) was released in 2008 and presented an open source opportunity for novice and expert users alike to readily harness the power of analysis and visualization tools with a focus on computer security. A few of the tools contained in the toolset had remained the same over the years but many had evolved with subsequent releases, thus causing DAVIX to become an outdated solution composed of these older versions. The need to provide a new version of DAVIX that incorporated the latest releases of the tools was paramount.

This project represents the full software development life cycle to produce a new version of DAVIX, rebuilt from the ground up. The user base was surveyed to identify current needs and wants in such a system. New requirements were established based on changing technology and a desire to ensure the solution did not stagnate in the future. New procedures for release and documentation were put into place and implementation began from scratch.

The only components retained from the original system were philosophical; the drive to provide the tools in the existing DAVIX such that functionality was not diminished, the goal of incorporating

new tools and technologies being used in the computer security visualization field, the intention that this solution would prove useful for a wide range of users, the accessibility of the solution based on its delivery method, and the focus on user documentation and help files that encouraged exploration and use of the tools.

For the sake of this document the original DAVIX (released in 2008) will be referred to as "the original DAVIX" and the new DAVIX (scheduled for release in May 2014) will be called by its version name of "DAVIX 2014".

Section 2 Problem Statement

The original DAVIX provided a valuable solution that integrates tools for data analysis and visualization in a single, usable toolset. The effort was spearheaded by Raffael Marty who envisioned the DAVIX toolset being both a learning tool for novice users and an all-in-one package for the expert that streamlined the data capture through visualization process. He tightly coupled the release of the original DAVIX with the publication of his book, Applied Security Visualization [3] and referenced DAVIX frequently in learning exercises throughout the text.

In addition to sponsoring the original DAVIX and authoring a book, Raffael Marty contributes significantly to the field through his ownership of the SecViz.org website [5] which provides a resource for the security visualization community in terms of posts about relevant technologies and events, and his participation in events and teaching workshops on the topic of security visualization.

Although the original DAVIX was developed on the open source model it has not been updated since its original released in 2008 and as a result some of the tools it contains are dated and new tools in the area of visualization have become prominent.

The original DAVIX was released in the Live CD format which allowed the user to burn the distribution to a CD and boot it directly without going through an installation process; this format has been a popular means of distribution in the Linux community since the late 20th century but it reduces the flexibility of the installation in that users cannot easily make modifications to the installed tools or otherwise customize the distribution for their own use.

In conjunction with this it makes the toolset itself more difficult to maintain and discourages regular releases to the user community. Finally, there has been a significant increase in the interest in virtual machines which is not wholly supported with a Live CD format.

A revised version of DAVIX that meets the changed needs in the computer security sector in terms of the tools provided and in the distribution methods while also embracing the collaborative principles of open source development makes this project of great value to the user community.

Section 3 Software Engineering Process

3.1 Methodology

There are a variety of software engineering methodologies and their successful application depends on an analysis of the project and the project stakeholders. A project may be best suited for a particular methodology based on the level of detail in the scope, the resoluteness of the stakeholders, the novelty and type of the application, the desired schedule, and other factors. In this case the methodology of this project changed during the design phase of the project as it became apparent that the initial approach would not meet project needs and stakeholder requirements. We began with the assumption that the scope was tightly defined, the stakeholder needs would not change, a strict deadline had been established. We had the additional assumption all team members would be actively available for the duration of the effort. As the project executed, we realized the methodology must change to ensure success.

3.1.1 Initial Approach

The project was initiated with a clear set of expectations for an end result, essentially a revamped version of the original DAVIX that incorporated the latest version of the existing DAVIX. A project plan

and scope of work was developed to account for all major components of a software development effort, from analysis through release to production with milestones concurrently supporting the stakeholder needs to have the new solution available for initial release at the May, 2014 Honeynet Project Workshop and for US release at BlackHat 2014.

The intention was to use a waterfall model of development with concise milestones and clear estimates for the amount of effort required from each resource by subtask. Expectations were established by stakeholders and although we were surveying the community for their needs, no risks were identified to indicate that we might need to modify the general tasks. We also believed that all project resources would have the availability to perform tasks as directed and within schedule.

Due to the structured nature of the work this approach seemed sensible and would ensure target dates were met and the project remained on schedule. To demonstrate this idealism, the following is the initial work breakdown structure produced in support of this effort.

#	Phase/Task
1	**Project Initiation**
1.1	Kick-off Meeting
1.2	Schedule/Project Planning
2	**Research**
2.1	Community Needs Survey, e.g., Survey Monkey or Poll
2.2	Review survey results.
2.3	Look into similar tools; e.g., Backtrack (Max Moser), Kali.org
2.4	Write up quick spec / approach
3	**Analysis**
3.1	Proposed Visualization Tools List, including licensing issues
3.2	List of all Tools/Modules included from existing version
3.3	Visualization Tool Prioritization (what goes on the final release)
3.4	Determine download area, user registration, etc.
3.5	What are infrastructure needs (e.g., apt repo)
3.6	Figure out licensing for re-packaging tools into a DEB
4	**Implementation**
4.1	Base OS
4.2	Application Packages
4.3	VM/Distribution Method
4.4	Build AWS Image
4.5	Build distro server
5	**Testing**
5.1	Test harness
5.2	Test cases
5.3	Run test cases; QA tasks
6	**Documentation**
6.1	Update DAVIX Documentation
6.2	Review/update tool specific documentation
7	**Release**
7.1	Update torrent
7.2	Update mirrors
7.3	Update Web site

Table 1 – Initial Work Breakdown Structure (June 2013)

As the project proceeded it quickly became apparent that much more flexibility in our approach was needed due to a number of factors. The primary issue affecting schedule adherence was that this project was indebted to the participation of volunteers.

3.1.2 Volunteer Participation

This project was developed under the open source model, which encourages the free distribution of software and the open participation of the community in the evolution and maintenance of the product. Community involvement can begin at any stage in an open source project, either from the very start or post-release. In the case of DAVIX 2014 we determined that including volunteers in the effort at an early phase would facilitate not only their interest in the project but also community understanding of our goals and a sense of ownership in the success of the project.

Raffael Marty was the project sponsor and primary stakeholder of the project as he was invested both professionally and personally in the success of DAVIX 2014. Volunteers were slated to work on the project throughout, as technical advisors, as graphic designers, and as developers. Their efforts were essential to the success of the project as the realistic effort and breadth of expertise necessary to

produce the solution within the required time frame was beyond the capacity of any single individual. The unanticipated result of using volunteers as project resources was their sporadic availability as they balanced other activities while contributing to the project. (This is elaborated on in section 4.2.1 - Lesson 1 – Managing a Team of Volunteers Requires Special Consideration as It Results In a Unique Project Team Dynamic.)

As a result we were forced to work the project schedule around the availability and skill sets of the volunteers. Situations arose where someone key to the implementation effort was unavailable for weeks at a time and the schedule required adjustment and other resources to fill those needs to accommodate it. In another case volunteers slated for the documentation effort were unavailable when the project demanded it and the task fell to the project manager. Related to this drive for flexibility, as more experts and stakeholders were identified and interviewed and the user community was survey, their needs and the associated requirements began to change.

3.1.3 Flux of Project Requirements

The project requirements were initially defined by the primary stakeholder, Raffael Marty. As we delved deeper into the project and

evaluated similar tools and the needs of the community at present we began to discover that those requirements were insufficiently scoped and that the needs were different and more substantial than we had anticipated. It became quickly apparent that as the project continued and new interest was generated in the community that we were receiving more feedback from experts that required us to proactively redefine our requirements as we went.

We could have easily continued as originally scoped but since the central tenant to the philosophy behind the tool (as described in Section 1 - Introduction) was that it be accessible and usable to the community, it would have been foolhardy to ignore the wisdom that was being imparted. These insights continued to be contributed throughout the duration of the project, requiring regular evaluation of our fundamental goals and identification of whether new suggestions ought to be implemented and the associated risk to the project in doing so. Due to the necessity for schedule flexibility and the ability to adapt to changing requirements, the methodology had to change.

3.1.4 Modified Approach

Agility then became the model for the project. There were still clearly defined major objectives with a specific deadline but an

elemental requirement to incorporate strong communication amongst the project team, regular re-evaluation of requirements, scope, and scheduling, and a constant call for new volunteers.

We pursued a pseudo-agile method in terms of encouraging communication and regularly evaluating requirements, but we held to a single development cycle, instead of iterative releases, with the intention of a first release that met the initial requirements and identified specific tasks for future releases.

3.1.4.1 Communication

Team communication in this project was facilitated with a number of tools, as it was a geographically distributed team. We shared spreadsheets and other project documentation using Google Docs, we conducted weekly status meetings over Skype, and copious emails were distributed with the understanding that all team members be involved in nearly every aspect of the project.

We had a core team which consisted of the project sponsor, my graduate advisor, the primary volunteer developer, and me. We would participate in weekly team meetings to update each other on the status of our tasks and ensure that any upcoming tasks were appropriately assigned. We would identify any issues that were being

encountered, including scheduling conflicts, resolve issues through discussion or identification of who would take on the task for additional research, and finally identify a complete task list for each participant moving forward.

In many cases due to schedule conflicts and prioritization issues it was not possible for one or more team members to attend the weekly meetings; in those situations we would do the best we could by checking in with the missing team members prior to or after the meeting and gathering their feedback independently. This was certainly a drawback to operating with a team that was unable to prioritize the DAVIX 2014 project when it came to more important work and personal obligations. This also greatly increased the amount of effort required of the project manager to ensure everyone was on task and aware of the general project status.

After each meeting, minutes were distributed with a focus on general project issues and requirement changes, along with clear identification of task responsibilities, action items, and expectations for the following week.

3.1.4.2 Requirements Evaluation

As part of our weekly status meetings new change requests would be considered. These would be addressed via discussion and decisions would be made on their inclusion based on input by the stakeholders, in terms of necessity and desire, and the project manager, in terms of its impact to the project schedule and goals. Existing requirements would also be evaluated in cases where scheduled resources became suddenly unavailable and their skill set was not easily replaced. In these situations existing requirements were discussed to determine how to reduce the project scope to accommodate the reduced availability of resources while staying within the timeline.

Concurrently, when the potential of reducing the scope came into question there would be a push to recruit additional volunteers with the goal of ultimately reinstating those dropped requirements. Even when volunteers were successfully recruited, having indicated both interest and availability, there was a significant management cost in terms of bringing them on the project team and getting them familiarized with the project, the communication process, their role, and the tasks assigned to them.

3.1.4.3 Quick Resource Onboarding

Volunteer management necessitated quick onboarding since the availability of many of the volunteers was temporary and subject to schedule restrictions. There was also significant risk that a volunteer would express interest and ultimately fail to deliver due to lack of interest, capabilities, or time. The project manager was responsible for initial communication with the volunteers, including a concise introduction to the project and a definition of their specific tasks based on their perceived skill set and interests. The project manager coordinated all volunteer activity and ensured access was granted as required, communication lines remained open, and evaluated their work product.

3.2 Requirements

We identified base requirements for the solution that emphasized the need for a distribution method that coincided with community needs, a set of current tools that supported security visualization, and user and technical documentation. The latter was a key component in the original DAVIX, greatly contributing to its usability for novice users while including the range of tools intrigued the experts. In addition, the recognition that DAVIX 2014 would need to

support community involvement and maintenance was paramount to avoid another six year gap between releases. The requirements defined in the following table summarize the aforementioned base requirements in addition to the desire for community involvement:

No.	Requirement	Description
1	Distribution Method	The solution must employ a distribution method that is accessible by novice users while encouraging flexible use and modification by expert users.
2	Tools (or Modules)	The solution must include tools that are widely used, stable, and innovative for the purposes of computer security visualization.
3	User Documentation	The solution must include user documentation for each visualization tool to provide an introduction to its use for novice users.
4	Technical Documentation	The solution must include technical documentation that describes the setup process for the distribution written in a clear and approachable manner.
5	Community Involvement	The solution must be available to the community for revision and a process for incorporating their changes into new releases must be established.

Table 1 – DAVIX 2014 Requirements

3.3 Analysis

The analysis phase began with gathering data on the original DAVIX and sharing them amongst the team using Google Docs. Subsequently this information was evaluated, the user community was surveyed, tool licenses were reviewed, a base development virtual machine (VM) was established, and a shared repository for release and documentation was created.

3.3.1 Initial Identification of Core Tools

An initial review of the tools contained in the original DAVIX was performed to establish a base list for inclusion in DAVIX 2014. There was a strong desire to not remove functionality unnecessarily so nearly every tool included in the original DAVIX would move on to the new solution, these would establish the core set of tools for DAVIX 2014 although candidate tools would later be evaluated for inclusion. As part of this effort the team familiarized themselves with the tools included in the package, particularly if their previous experience was limited. Nine of the tool examples I created can be found in the appendix of this document beginning with Appendix C - Tool Example – InetVis; the tools evaluated include InetVis, OpenHeatMap, PADS, Afterglow, GraphViz, rumint, Timesearcher 1, Treemap, Tulip, Walrus, and Wireshark with some examples utilizing several tools.

As part of the identification effort, the versions of the tools in the original DAVIX were compared to the latest stable and latest beta versions released. This provided insight into which tools had remained relatively static over the years and which were actively modified. Tools that were identified as static were drawn into DAVIX 2014 as they were and tools that had been evolving over the years

were more closely scrutinized in terms of changes to their functionality and whether they would be an asset to the new solution.

This comparison also provided us with additional information when we began implementation and subsequently testing of DAVIX 2014 since the frequently changing tools were more likely to display stability issues and we had to test their new and changed functionality.

3.3.2 Community Survey

Community investment in the project was important so one of our first steps was to create a survey using Survey Monkey and publicize its availability to the user community. The DAVIX Release Survey [4] focused on identifying the actual user base and clarifying their wants and needs for the new solution in terms of the distribution method and tools for inclusion. The survey was publicized via the SecViz website [5] and Twitter and received over 90 responses, the majority within the first month of release. The survey results are available in Appendix B - Survey Monkey Results.

These survey results were used in identifying requirements going forward and also helped evaluate the general expertise and type of user who might use DAVIX 2014.

3.3.2.1 Survey Questions

The following are the survey questions distributed to the community and our motivation for each question. A complete list of questions and answers can be found in Appendix A – Survey Questions and results are located in Appendix B - Survey Monkey Results.

Our primary rationales for the survey were to learn more about the user community, including their skill set, their occupation, and their familiarity with DAVIX and the tools. We also were looking for information on community needs and desires for DAVIX 2014 in terms of the delivery method and the tools for inclusion. This provided us with information on the types of data being analyzed and the tools that the community found valuable and allowed us to use the survey results when evaluating candidate tools in a later phase.

No.	Question	Rationale
1	Are you or have you been a DAVIX User?	Identifies the user community.
2	How would you like to see DAVIX delivered?	Identifies the desired delivery method for DAVIX 2014.
3	What kind of data do you mainly work with?	Identifies the user community.
4	Are you a … (e.g., security analyst)	Identifies the user community.
5	How do you intend to use or are already using DAVIX?	Identifies the desired functionality for DAVIX 2014.
6	Are you fluent in UNIX?	Identifies the user community.
7	What (security) visualization tools are you using?	Identifies the desired functionality for DAVIX 2014.
8	If we made it easy for you, would you want a Web service environment to use Web	Identifies the desired functionality for DAVIX 2014.

No.	Question	Rationale
	visualization libraries on DAVIX?	
9	What are your biggest challenges when analyzing and visualizing security data?	Identifies the user community and desired functionality for DAVIX 2014.
10	Any other tools you need or want to see included in DAVIX? Is there any other input you would like to give us for the upcoming DAVIX release? Speak up now!	Identifies the desired functionality for DAVIX 2014.

Table 2 – DAVIX Survey Questions and Rationale

3.3.3 Candidate Tool Identification

New tools were identified by the project team and evaluated based on their application for security visualization, their stability, their ease of use, and community interest in the tool as identified through the aforementioned survey. New tools included Google Earth, FlowTag, dns-browse, netsed, nsm-console, PRADS, R Studio, rsyslog, and tcpstat.

Although the focus of DAVIX 2014 is on the visualization tools, the capture and processing of data is equally as important to support visualization tasks and a few new tools in those categories were also included. The reviews conducted were informal but tools were submitted to the project stakeholders and project manager prior to official inclusion. A comprehensive tool list has been outlined in section 4.1.4 - Tools Included.

3.3.4 Review of Tool Licensing

Although many of the tools fell under General Public Licenses some required review of the specifics of the license or written permission from the owners to ensure DAVIX 2014 would be legally compliant with the licensing requirements for its inclusive tools and public distribution. The list of tools was reviewed to confirm no issues existed and that any previously granted permission was still applicable.

3.3.5 Base VM Identification

A base VM was necessary to provide team members that were not directly involved in the development effort the opportunity to access the solution for the purposes of writing documentation and testing. The latest development version would be deployed to the VM and then it would be shared by the project team. During a team meeting we decided that this VM was best hosted by UAF's RAVE lab due to team familiarity with the technology and our access UAF resources for the completion of this graduate project.

3.3.6 Repository

A team repository was required to store and share source code during the implementation effort while tracking changes made by

different team members. After some discussion GitHub was selected due to it being a versatile and accessible repository that could provide source control during the development effort prior to release, in addition to managing source control for subsequent community involvement. It also conveniently provided a means of document management for technical documentation and user guides via its wiki feature.

A major benefit to GitHub was that we could allow any GitHub user to contribute to the project, thus supporting our goal to make DAVIX 2014 and its subsequent maintenance a community effort. As a result the DAVIX GitHub repository [1] was created along with the associated DAVIX Wiki [2].

3.4 Implementation

Implementation began by the establishment of common development environments for all involved in the engineering of the final solution. A repository, previously identified for hosting on GitHub, was created and team members were given access. Actual implementation of scripts occurred concurrently with a variety of other related tasks such as branding and documentation.

3.4.1 Development Environment Setup

Developers established their own development environments although it was initially agreed upon that Ubuntu 12.04, 64-bit, would be as the base platform. Subsequent issues during the implementation process with certain tools not being stable on the 64-bit platform led to an ultimate switch to a 32-bit platform (Ubuntu 13.10) and the decision to make conversion to 64-bit an issue to be addressed in the third release as described in section 4.3.2 - Plans for Third Release.

3.4.2 Base VM Setup

A base VM was also created to facilitate revision to the menu XML files concurrent to actual development of the scripts and distribution. This was hosted by UAF's RAVE lab. This VM was also used as a reference point to revise user documentation and as an easily accessible portal for team members indirectly involved in the project who wanted a glimpse as it progressed.

3.4.3 Repository Setup

A DAVIX repository [1] was established on GitHub for purposes of code management and wiki use. Team members were all provided

administrative access to the project and the DAVIX Wiki [2] was set to editable by the GitHub user base.

3.4.4 ISO Image

Work initially began on the ISO image distribution to provide a proof of concept that the base platform, tools and all other aspects provided a cohesive solution. This also created the first version of the image used for testing and documentation.

The distribution itself was based on Ubuntu and each tool identified for inclusion in DAVIX 2014 was carefully loaded into the development environment then tested. This was where stability issues with the 64-bit platform were discovered and a decision to revert to 32-bit was made. Details on other aspects of the implementation process are described in later sections.

3.4.5 Scripting

In addition to the ISO image, scripts were created to automate the fetching of packages and dependencies into a pre-established base environment. This would provide the basis for collaboration as expert users could take these scripts, run them against their environment, and make modifications as necessary to fine tune the DAVIX 2014 distribution to their specific needs. If, in the process of

this tuning, they discovered better ways of doing things or additional tools to add that did not negative impact stability then they might push these changes up to the DAVIX GitHub repository [1].

3.4.6 Testing

Informal testing occurred throughout the implementation effort by the developers and by technically adept team members. This included accessing the latest development instance of DAVIX 2014 on the VM, confirming the required tools were in the distribution, and verifying that they could be successfully run. More elaborate testing occurred post-implementation as described in section 3.6 - Testing.

3.4.7 Logo

The creation of a new logo to reflect DAVIX 2014 was another part of the process. The intention was to rebrand the solution to emphasize it had been updated in all ways and was no longer dated. Although a number of volunteer graphic designers were solicited this proved to be a challenging process due to the subjective nature of visual elements, particularly one that would be so vital and conspicuous as a logo. For reference purposes, the original DAVIX logo:

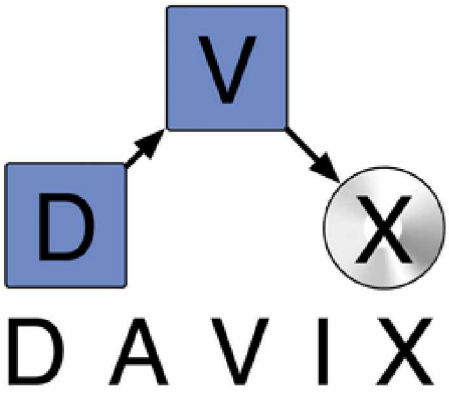


Figure 1 – Original DAVIX Logo

This clearly had an emphasis on the Live CD distribution method which would be an inaccurate portrayal of the DAVIX 2014 distribution methods. It was also somewhat dated in design.

After a number of iterations and reviews by the project team, DAVIX 2014 adopted a two-logo solution, one for black backgrounds and one for white backgrounds. This allowed greater adaptability to users with a strong preference for inverted color schemes. The emphasis of the logo was on the linked graph and specialization of visualization in the solution.

Figure 2 – DAVIX 2014 Logo on White

Figure 3 – DAVIX 2014 Logo on Black

3.4.8 Version Name

The name for the next version was subject to debate. We began the project referring to it as DAVIX NG but when we began

discussing the process of branding the distribution we ultimately chose the name by polling the project team with the options of DAVIX NG, DAVIX 2, and DAVIX 2014. The last democratically decided upon as ideal as future releases could reference future years and the age of the solution would be immediately apparent.

3.4.9 Branding

Branding the ISO image distribution involved setting up the desktop to feature the DAVIX 2014 logo, setting the color scheme of the desktop, and configuring toolbars to customize the interface and encourage clear associations with the DAVIX 2014 brand.

3.4.10 Menus

The original DAVIX contained menu items to some standard tools such as Firefox, which included a set of DAVIX specific bookmarks, in addition to three primary menus to organize the three types of tools – Capture, Process, and Visualize. A menu item pointing to a tool was contained in its own submenu such that a menu item for the individual user guide for that tool was also readily available. The original DAVIX also employed quick start guides so that menu items created for command line tools would open a

terminal, displaying quick start text (pulled from a free standing text file) and then a command prompt.

DAVIX 2014 required a maintainable solution when it came to documentation, as mentioned later in section 3.5 - Documentation, thus all documentation was placed in the DAVIX Wiki [2]. This meant menus were created with a similar structure to that of the original DAVIX, such that each tool had a submenu with then a menu item for the tool and then a menu item pointing to the user guide. The latter was a link to the user guide for that tool on the DAVIX Wiki [2] which introduced a new complexity as linking to externally hosted information would defeat the goal of an inclusive, standalone solution.

3.4.11 Static Wiki

Due to the desire to support a wholly self-contained solution, it was determined that with each release the DAVIX Wiki [2] would be pulled down as a set of static HTML pages and packaged into the ISO image. This would provide a set of documentation within the solution current for the tools in that release. Meanwhile it allowed individuals running DAVIX 2014 via scripts to access the dynamic DAVIX Wiki [2] via the GitHub repository for the latest information.

3.5 Documentation

A comprehensive set of user guides and technical documentation were included in the DAVIX 2014 release with the intention of having community involvement in refining them and making them more helpful over time.

3.5.1 User Guides

The original DAVIX was released with multiple types of user documentation; "Quick Start" text files for all command line tools, individual PDF files for each tool to be presented in the menu, and a comprehensive "DAVIX User Manual" [6] that included setup and use of the Live CD along with every user guide for the individual tools.

As noted previously this was certainly not easily updated or maintainable. It would not fall in line with the philosophy of encouraging collaboration and participation by the community so we chose to use a wiki to support all user guides and documentation. The DAVIX GitHub repository [1] established for code management already featured a wiki so it was a matter of adding user guides for each of the DAVIX tools. A general template was established featuring the name, version, site, summary, links to comprehensive user manuals and other documents, a command line quick start

section (if applicable for the tool), and a section detailing how to get started with the tool. An example of a user guide on the DAVIX Wiki [2] has been provided below:

User Guide: AfterGlow

Edit Page | Page History | Clone URL

Site

afterglow.sourceforge.net

Summary

- Tool to convert CSV input to a DOT graph description. AfterGlow takes a configuration file that configures how the nodes and edges are represented in the DOT file. The DOT file can then be graphed via Graphviz.
- In addition to the main tool, AfterGlow ships a set of tools to convert CSV data into data formats that can be used with other visualization tools.
- Includes capper.pl script from Raffael Marty's book "Applied Security Visualization".

Links

AfterGlow 1.x Manual

Command Line Quick Start

```
Example:        echo -e "a,b\nc,d\nc,e" > test.csv
                cat test.csv | afterglow.pl | neato -Tgif -o test.gif

Manual:         afterglow.pl -h
```

How to Get it Working

- Open a console.
- First a CSV file of sniffed network traffic has to be generated using the command:

```
tcpdump -vttttnneli eth0 | tcpdump2csv.pl "sip dip dport" > sniff.csv
```

- Open Firefox and do some extended surfing.
- Press Ctrl-C in the console window where tcpdump is running.
- To transform the CSV file to a GraphViz dot file execute:

```
cat sniff.csv | afterglow.pl > sniff.dot
```

- To render the sniff.dot into a GIF file use the command:

```
neato -Tpng -o sniff.png sniff.dot
```

- To view the result open GQview with command:

```
gqview
```

Comments

Last edited by AmandaLG, 9 days ago

Figure 4 – DAVIX 2014 Wiki – User Guide: Afterglow – Example

The process of creating the documentation involved the initial conversion of existing user guides to the new wiki format, review of

the versions, links and updating the directions themselves to correlate with the changes to installation paths. New tools were then added to the DAVIX Wiki [2] using the same template, albeit only the visualization tools have been provided with "How to Get it Working" sections in this release. Additional user guide documentation for the capturing and processing tools was deemed a very low priority by the project sponsor and out of scope for the initial release.

The benefit to the wiki approach is that contributors can easily modify the DAVIX Wiki [2] if they notice errors and expound on any information provided. Then when a new release of DAVIX 2014 is prepared the DAVIX Wiki [2] will be drawn down into static files and packaged into it, thus capturing the contributions of the community not only to the release itself but to the documentation.

3.5.2 Technical Documentation

The original DAVIX had a comprehensive user manual [6] contained in a PDF file but since DAVIX 2014 was moving in the direction of shared responsibility for maintenance of documentation it was necessary to host the technical documentation on the aforementioned wiki as well.

3.5.2.1 Setup Process

Technical document was required to detail the setup process for the various distribution methods; a short version was presented as a text file in the primary directory of the DAVIX GitHub repository [1] with more detailed information available in the DAVIX Wiki [2]. The "Setup Process" wiki page described the two distribution methods (scripts for use or modification and the ISO Image) and then outlined how to get each working. Dependencies and any environmental requirements were specified, along with a step by step process for running DAVIX 2014.

3.5.2.2 Contributor Process

Although outside the scope of this initial release, documentation will need to be produced to provide a basic guideline for contributing to future releases by way of script modification and the business process that would occur prior to an official release to ensure quality assurance measures were met. This page has been stubbed into the DAVIX Wiki [2].

3.6 Testing

The quality assurance (QA) process was the shared responsibility of the entire project team although communication and prioritization

of QA tasks were coordinated by the project manager. This release of DAVIX 2014 was tested manually; no automated tests were established as they were out of scope for the initial release (see section 3.6.1 - Testing Harness). Due to all testing being performed manually we were forced to rely on the diligence of our volunteer testers and this presented a problem in itself as some were less attentive to issues or lacked the basic experience with Linux that we expect of our users, thus increasing the time we had to spend performing quality assurance management.

3.6.1 Testing Harness

Although the establishment of a test harness to ensure script quality was proposed to be part of this release in the initial project plan, this was ultimately reprioritized based on other requirements and limited resource availability. It has been moved into the second release, as described in section 4.3.1 - Plans for Second Release, and will be used to support contributor maintenance of the scripts.

3.6.2 Scripts

Once the initial scripts were created, the entire project team was tasked with review and testing to ensure that the tools would install and run seamlessly for the user community. The set of scripts would

fetch the tools from the Ubuntu Advanced Packaging Tool (APT) or, if appropriate, other locations, and install them in the environment on which they were run. Our first step was a visual review of the scripts which produced significant feedback on ensuring checks were performed for the platform and discussions on whether to specify version numbers for packages or allow the scripts to get the latest versions from the Ubuntu APT. This is also indicative of how the scope continued being refined through the project. The decision was made to provide two versions of the script, to be provided in the second release, as described in section 4.3.1 - Plans for Second Release.

Subsequent to the visual review we each began with the same environment, Ubuntu 13.10 32-bit, and started testing the scripts themselves, debugging the scripts as we went and pushing updates to the DAVIX GitHub repository [1]. Debugging the scripts involved confirming there were no errors when they ran and if a version for a tool was specified that it coincided with the version in the list of required tools.

3.6.3 ISO Image

The ISO image was similarly tested by the project team; first ensuring we could load it onto a VM and then running through all of the solution components, including verification of the following:

- That each menu item was correctly linked;
- The associated static wiki was available;
- DAVIX related bookmarks were in Firefox; and
- Each tool ran correctly from the menu.

3.7 Release

As the solution was being released on GitHub the final deployment was simply ensuring the latest scripts and image were available in the repository, a torrent was available with the ISO image, and then notifying the user community of the new release.

3.7.1 GitHub Push

The final DAVIX 2014 scripts were pushed to the DAVIX GitHub repository [1]. As all documentation had been modified within the DAVIX Wiki [2] there was no official release of the associated documentation; the final versions were already there.

3.7.2 Torrent and Mirror Update

A torrent and mirrors were used for distribution of the ISO image; which was the same technique used for the original DAVIX Live CD distribution.

3.7.3 Release Plan

The final step to release will occur during the Honeynet Project Security Workshop in Warsaw, Poland in May 2014. DAVIX 2014 will be officially launched at the workshop, which will involve notifying the community of the updates; rebranding the DAVIX site (located at secviz.org/node/89) with the new logo and providing information that directs users to the DAVIX GitHub repository [1] and torrents. The community will also be notified via a post to the SecViz site and Twitter. A secondary U.S. release will occur at BlackHat 2014 in July.

Section 4 Conclusions

4.1 Final Product

The final product, DAVIX 2014, will be released in May 2014 and will incorporate the functionality of the original DAVIX along with several new tools and more importantly an interface for community involvement in the solution as it evolves.

4.1.1 Repository

The requirements for this project associated with the repository were successfully met. The DAVIX GitHub repository [1] will host both one of the DAVIX 2014 distribution methods and comprehensive technical and user documentation on the solution. This will provide members of the community with access to the toolset for use and establishes a baseline for contribution. Individuals can opt to act solely as users by downloading the distribution and viewing the documentation or they may begin contributing to the project by making changes to the technical and user documentation in the DAVIX Wiki [2].

4.1.2 SecViz Web Site

The SecViz site [5] will host the main DAVIX page with links to the DAVIX GitHub repository [1], links to the torrent to download the

ISO image, along with other information on the release. This successfully meets the need for publicizing the release of DAVIX 2014 and providing a web site devoted to the solution. It also fulfills the requirement to effectively distribute the ISO image.

4.1.3 Distribution Methods

The final DAVIX 2014 solution will be released via a GitHub repository and feature several methods of distribution, both a set of scripts a user can run against their installation of Ubuntu and an ISO image. This increases the flexibility of the solution as the technically savvy individuals can easily modify them for their own purposes without much effort while less experienced users can easily access all of the tools by merely loading the image onto a VM. This meets the requirement of ensuring DAVIX 2014 employs an accessible distribution method for novice and expert users alike.

4.1.4 Tools Included

DAVIX 2014 includes a total of 55 tools with 46 being the most recent versions of tools that were included in the original DAVIX, along with 9 tools new to DAVIX 2014. Some of the tools fulfill multiple roles as such there are 20 tools that involve capturing data, 11 tools that process data, and 28 tools for data visualization. This

meets the requirement to ensure none of the functionality existing in the original DAVIX was lost while providing an improved solution via the latest releases of legacy tools and the addition of new tools that have become valuable in the security visualization field. The following lists all of the tools included in this release of DAVIX 2014:

Tool	New to 2014?	Capture	Process	Visualize
Afterglow	No			
Argus	No			
BroIDS	No			
Chaosreader	No			
Cytoscape	No			
dns-browse	Yes			
dnstop	No			
EtherApe	No			
FlowTag	Yes			
GeoIP	No			
GGobi	No			
glTail	No			
GNUplot	No			
Google Earth	Yes			
Graphviz	No			
GUESS	No			
InetVis	No			
Mondrian	No			
MRTG	No			
netsed	Yes			

Tool	New to 2014?	Capture	Process	Visualize
Nfdump	No			
Ngrep	No			
Nmap	No			
nsm-console	Yes			
NVisionIP	No			
Octave	No			
p0f	No			
PADS	No			
Parvis	No			
Ploticus	No			
PADS	No			
PRADS	Yes			
Processing	No			
R Project	No			
R Studio	Yes			
RRDTool	No			
RT Graph 3D	No			
Rumint	No			
Rsyslog	Yes			
Scapy	No			
Shoki	No			
Snort	No			
syslog-ng	No			
Tcpdump	No			
Tcpflow	No			
Tcpreplay	No			
Tcpslice	No			

Tool	New to 2014?	Capture	Process	Visualize
Tcpstat	Yes			
Tcpxtract	No			
Timesearcher 1	No			
Tnv	No			
Treemap	No			
Tulip	No			
Walrus	No			
Wireshark	No			

Table 3 – DAVIX 2014 Tools

4.1.5 Documentation

Documentation is maintained in the GitHub DAVIX Wiki [2] along with for a static copy of the DAVIX 2014 Setup and User Guides in the ISO image. This successfully provides accessible and current access to the documentation online, along with permitting the community to add to that documentation. It also provides a static inclusive set of documentation in the ISO image for situations where Internet connectivity is unavailable.

4.2 Lessons Learned

4.2.1 Lesson 1 – Managing a Team of Volunteers Requires Special Consideration as It Results In a Unique Project Team Dynamic

The project team consisted entirely of volunteers which made their coordination and availability a challenge compared to a more

traditional approach of paid employees or contractors. The only motivation for volunteers to participate in the project consisted of adding to their technical experience and, more importantly, the ability to participate in the early stages of the project and ideally guide it in the direction they were most eager to see it grow. The majority of the volunteers were interested in DAVIX 2014 for their own benefit in terms of being able to provide input into a solution they anticipated using after its release.

The project was fortunate to benefit from steady participation from a core group of members in addition to an influx (and subsequent departure) of other resources. This also introduced substantial issues in terms of successfully managing the project.

Part 1 – Volunteers Are Inconsistently Available Due To Prioritization of Work and Personal Life

The majority of the volunteers were contributing to the project while balancing work and personal lives. This meant we had a lot of issues where volunteers working on integral portions of the project would suddenly deprioritize the DAVIX 2014 effort in favor of their other obligations and we would have to scramble to find someone who could complete the task in their place. This emphasized the importance of keeping code, documentation and all resources

communal so that new resources could more easily pick up tasks that were left abruptly. Nonetheless, this was a significant challenge particularly when availability issues could not even be planned on.

Similarly, some volunteers committed months earlier to participate but were unavailable when the time for their tasks arose. As a result of these issues we made an effort to rely as much as possible on duplication of skills amongst team members and contingency plans for missing team members.

Part 2 – Volunteers May Be Eager To Participate But Lack the Necessary Qualifications to Efficiently Contribute

Since DAVIX 2014 was a volunteer project we welcomed all who were interested in dedicating their time and knowledge to the solution. In some cases this meant people with limited experience in certain areas would undertake complex tasks in unfamiliar technologies and learn as they went along. As a result initial outputs would often need substantial review and revision by more expert team members.

In most situations this was acceptable since a certain degree of peer review is expected, although the degree to which revisions were required was often unanticipated and added to the project scope and necessary investment by the subject matter experts. That said, many

volunteers who were working beyond the scope of their capabilities required consistent oversight and review to ensure their work product was sufficient, thus each volunteer was evaluated by management to determine whether their contributions outweighed the expense in time and energy of other team members.

In a few cases we had volunteers who were completely inexperienced and whose participation served as a detriment to the project as they lacked the initiative or skills essential for even basic tasks. These individuals required more up-front management before their inadequacies were identified and as a result extra effort had to go into the reversion of any issues they introduced.

Part 3 – Managing Volunteers Can Be Very Time Consuming Due To Their Personalities, Expertise, and Availability

Management of the volunteers in itself offered its own set of difficulties. Status meetings were often not attended due to scheduling conflicts, depending on the skill set of the individual instructions and onboarding, as mentioned in section 3.1.4.3 - Quick Resource Onboarding, could be a time consuming process. Finally, we had one instance of a volunteer whose productivity was vastly outweighed by the amount of effort expended by the project team in review and management of their work product. In this situation the

volunteer was allocated to several different tasks before we identified there was no niche they could fill with minimal supervision. In this situation the lesson was again emphasizing team cooperation at all levels and quickly identifying volunteers who could not productively provide assistance.

4.2.2 Lesson 2 - The Scope of a Project Will Change

Part 1 – Requirements Are Fluid When the Project Scope Is Not Well Defined

With regards to scope a very valuable lesson was learned in terms of allowing fluidity of requirements particularly in light of such a loosely structured project team and inconsistent communication. This pairing of changing scope along with volunteer availability increased the project schedule substantially. Due to the desire of the project sponsor, Raffael Marty, to see DAVIX 2014 launched at the Honeynet Project Security Workshop, May 2014, some of the requirements were evaluated by the core project team and deemed a lower priority, thus suitable for a future release. In retrospect either the volunteers or requirements would need to be defined and relatively fixed or the project schedule would need to be set organically and exist just as fluidly as the shifts in the team and scope.

Were I to repeat this project I would more tightly manage requirements because that would be easier to do than locating volunteers with the appropriate skill set and availability to devote themselves to the lifespan of the projects.

Part 2 – The Project Scope Will Increase When the Project Sponsor Is Professionally Invested In the Outcome

Raffael Marty, our project sponsor and primary stakeholder, was invested in DAVIX 2014 since his professional reputation was at stake and he was extremely involved in ensuring the project proceeded according to his vision. This resulted in a lot of scope creep as his frequently changing opinions could not be discounted since the release was dependent on his approval. If he had a strong vision then we were ultimately forced to comply even if this meant an increase in scope. If this were a contracted project, involving a monetary exchange, then it may have been easier to keep his expectations in check due to the nature of defining the project and controlling for change.

Part 3 – The Project Scope Will Increase When Stakeholder Expectations Are Not Kept In Check

One of the primary sources of scope creep and changing requirements were feedback from stakeholders and volunteers. A

team member would hear about a new technology or remember a useful feature and propose it to the group. Or, in some situations, proceed to spend hours implementing it without running it past the team first. In some cases this type of initiative and enthusiasm was very beneficial to the project but in others it led to dissatisfaction if a particular desire was determined to be out of scope, or if the work product ended up destabilized as a result of the new feature the team member decided to incorporate. The latter case would then add to overall effort to resolve the issue by either moving forward or back tracking, not to mention the original time that was lost in implementing something out of scope.

4.2.3 Lesson 3 – Quality Assurance Is a Critical Element to Software Development

The value of quality assurance cannot be underemphasized. There were two aspects to QA that were lessons learned in this project; these issues were peer review and overall product testing.

Part 1 - Peer and Expert Review Is Essential When Working With Volunteers of Varying Skill Sets

We discovered quickly that peer and expert review of work products was essential to ensuring a stable release version. The initial supposition that volunteers were independent and capable in

the required skill set was quickly set on its head when glaring issues were discovered in deliverables. As mentioned previously this added substantial time to the work load of subject matter experts as they were called upon to review and revise these deliverables. It also emphasized the value of a core team of experts who were available on an as needed basis for this sort of task. In the future I would attempt to offset some of the need for peer review by initiating these reviews earlier in the project and ensuring issues are communicated and addressed sooner rather than later.

Part 2 – Product Testing Must Be Performed Throughout the Project to Ensure A Stable Release Version

DAVIX 2014 would have greatly benefited from automated testing and a dedicated team of volunteers for testing purposes. Due to a lack of skilled resources we did not have a team of devoted testers. We also decided automated testing was a low priority and relegated it to a future release (see section 3.6.1 - Testing Harness) which in retrospect was not an ideal choice. Were this project to be repeated a greater emphasis on QA would be placed and developer work would be managed with the expectation of simultaneous implementation of automated testing. A greater effort to rally the user community for testing prior to release would also be planned

for, perhaps even beginning with collecting volunteers at the start of the project when the community was surveyed.

4.3 Future Enhancements

During this process a number of software enhancements were identified which were outside of the scope of the original project. Due to the importance of scope management and ensuring the project was completed within schedule for the purposes of this graduate project, these issues were logged and identified as future enhancements. Each of these enhancements went through an informal change control process wherein the core project team (including the project sponsor and project manager) acted as the change control board to identify features that were not essential for the initial release but could be effectively addressed in the future.

Subsequent releases will be handled by DAVIX project team and any community members who wish to participate, but a rough roadmap for the future of the project has been created as a result of this effort. There are plans for a second release of DAVIX 2014 which will feature several enhancements, including a testing harness to ensure community updates to the scripts are appropriately vetted

and the hosting of Debian packages that are not pulled using Ubuntu's APT.

4.3.1 Plans for Second Release

The next release will incorporate some of the requirements that were previously deemed low priority and not incorporated in the initial release, including hosted Debian packages, a testing harness for test automation, and multiple versions of the scripts to allow users to elect to get the latest versions of the tools or the latest stable versions of the tools.

4.3.1.1 Hosted Debian Packages

Some packages were not pulled using Ubuntu's APT but were installed manually via script. Since the location of these packages is subject to change without notice the scripts would then be useless. A solution is to build each package and host it in the DAVIX GitHub repository [1], and then point all scripts to those versions of the packages. This would require some maintenance if a new version of a package were released, since its associated copy in the DAVIX repository would need to be updated. At the same time it would eliminate the risk associated with pointing to packages that may be moved without notice in the future.

4.3.1.2 Testing Harness

As mentioned previously, the testing harness was identified as an enhancement for a future release. This will be essential for implementing contribution by the user base since it will facilitate QA prior to release. This is also slated for the second release of DAVIX 2014.

4.3.1.3 Two versions of scripts (stable and latest)

When we initially reviewed the DAVIX 2014 build scripts there was an immediate debate as to whether they should fetch the latest versions from APT or if version numbers ought to be specified. The benefits of the former were ensuring when the scripts were run that the latest versions were used but the consequences were the potential of impacting the stability of other tools if they were not thoroughly tested. The benefits to clearly specifying the version numbers were that QA measures could establish the script was stable but with the unfortunate side effect that the scripts would require manual maintenance on a regular basis whenever versions of tools in APT changed.

Consequently it was determined that the initial release of DAVIX 2014 would feature a script fetching only the latest versions but that

the second release would have two scripts to accommodate the risk takers and the more cautious users. It was informally decided that if contributions were not actively made to the versioned script after some length of time then it may be pulled from the project completely.

4.3.1.4 Enhancements to documentation

The documentation, both user guides and technical documentation, is an evolving process. The initial release only featured functional working guides for the visualization components of the tools but a future release could include similar guides for the capturing and processing tools as well.

4.3.2 Plans for Third Release

4.3.2.1 64-bit support

As the tools were developed independently according to the objectives of their developers, several tools behaved poorly in the 64-bit environment and stability was compromised. Ultimately a 64-bit environment is desirable for larger memory stores and the processing of higher quantities of data. A 32-bit environment only allows the use of up to 4GB of memory which restricts the efficiency of processing and its ability to handle extremely large sets of data.

This subsequently has an impact on the visualization of large sets of data or complex data sets since they would need to be broken down into smaller components to run in a 32-bit environment.

In order to operate effectively in a 64-bit environment all of the tools have to be native to that environment. A 32-bit tool running in a 64-bit environment will not harness the efficiency that the environment itself offers with regards to memory usage. This is also a key concern of the user community who may want to process large sets of data efficiently and may already have established 64-bit environments, thus a 32-bit version of DAVIX 2014 would be undesirable. As such debugging the problem tools and converting DAVIX 2014 to 64-bit support is slated for the third release.

4.3.2.2 Quick Start Menu Support for Command Line Tools

Although the original DAVIX release featured menu items for command line tools that opened a terminal window, displayed a "Quick Start Guide" and then the command prompt, this was out of the scope of DAVIX 2014. In the third release might be desirable to reincorporate this feature to increase the support of users new to the tools.

4.3.3 Long-term Vision

4.3.3.1 DAVIX Debian Package

Ultimately the DAVIX GitHub repository [1] would also include a Debian package for DAVIX incorporating all of its dependencies. This single package would include the DAVIX build scripts along with all of the hosted Debian packages (implemented in the second release) for modules that are not maintained by Ubuntu APT, in essence providing a new type of distribution of DAVIX which is all-inclusive, much like the ISO image, yet geared towards expert users in that the scripts can be customized.

4.3.3.2 Streamline usability of log files

Another future revision is to streamline the usability of logs by harnessing log repository tools that store captured data in a single location. The tools in DAVIX 2014 would then be directed to default to that log location for processing and visualization of results, hence reducing the amount of time the user has to spend storing and relocating log files in the system.

4.4 Unexpected Outcomes

At the inception of the project, interest was expressed by Jaguar Land Rover (JLR) to expand on DAVIX 2014 and use it for data

analysis and visualization of real-time automotive data in their future Infotainment systems. JLR subsequently became involved as a potential stakeholder in the project with the intention to evaluate it for modification and use. Having a commercial entity interesting in open source software collaboration would prove to be beneficial to DAVIX 2014 and its future.

JLR is currently focused on modifying the solution to work in a 64-bit environment since it will be essential for their goals of processing large matrices of data. If they arrive on a solution and contribute to the project then a 64-bit DAVIX 2014 may not necessarily wait for a third release but could be pushed forward more quickly.

Section 5 Glossary

APT

Ubuntu's Advanced Packaging Tool provides access to the core libraries available for installation into the Ubuntu operating system.

DAVIX

A data analysis and visualization Linux toolset incorporating modules useful in the capturing, processing, and visualization of computer security data.

ISO Image

A disk image with a file extension of .iso.

Live CD

A bootable installation complete with operating system that runs on memory.

VM

A virtual machine is a software emulation of a machine from its system configuration to its operating system and installed software.

Section 6 References

[1] A. Gellhouse, DAVIX GitHub Repository. 2014. Available: https://github.com/secviz/davix

[2] A. Gellhouse, Data Analysis and Visualization Linux Toolset Wiki. 2014. Available: https://github.com/secviz/davix/wiki

[3] R. Marty, Applied Security Visualization. Upper Saddle River, NJ: Addison-Wesley Professional, 2008.

[4] R. Marty and A. Gellhouse, DAVIX Release Survey. 2013. Available: http://www.surveymonkey.com/s/769KG3C

[5] R. Marty, SecViz Website. Available: http://www.secviz.org/

[6] J.P. Monsch and R. Marty, DAVIX Version 1.0.1 User Manual. 2008.

Appendix A – Survey Questions

The following are the survey questions and answers as posed in the survey. Questions with no defined answers are open text fields. Bulleted lists indicate multiple selections are permissible and lettered lists indicate only a single option may be selected. All questions could be skipped.

1. Are you or have you been a DAVIX User?
    a. Yes
    b. No
2. How would you like to see DAVIX delivered?
    - AWS Image
    - VM Image
    - ISO Image (CD)
    - Other (please specify)
3. What kind of data do you mainly work with?
    - NetFlow or other traffic flows
    - PCAP (raw packet captures)
    - Firewall

- IDS / IPS
- Host logs from UNIX
- Host logs from Windows
- Proxy logs
- Web logs
- Application logs
- Other (please specify)

4. Are you a ...
    - system administrator
    - malware reverse engineer
    - security analyst
    - developer/programmer
    - forensic expert
    - incident responder
    - security engineer
    - security architect
    - security consultant

- Other (please specify)

5. How do you intend to use or are already using DAVIX?

    - As a log centralization and analysis workstation (using rsyslog, syslog-ng, logstash, or other to collect data and then analyze it)
    - As a forensic log analysis distro (we get logs, load them on DAVIX and analyze them there)
    - To try out visualization tools without having to install them myself
    - Other (please specify)

6. Are you fluent in UNIX?

    a. Yes

    b. No

7. What (security) visualization tools are you using?

    - AfterGlow
    - AfterGlow Cloud
    - AfterGlow for Splunk
    - GraphViz
    - GUESS
    - InetVis
    - Large Graph
    - Ploticus
    - Processing
    - R Project

- ChartDirector
- Cytoscape
- EtherApe
- Gephi
- GGobi
- ggplot2
- glTail
- GnuPlot
- Layout (LGL)
- Maltego
- Mondrian
- MRTG
- NFSen
- NVisionIP
- Parvis
- PicViz
- RRDtool
- RT Graph 3D
- rumint
- Timesearcher 1
- tnv
- TreeMap
- Tulip
- Walrus
- Commercial Visualization Tool or Other (please specify)

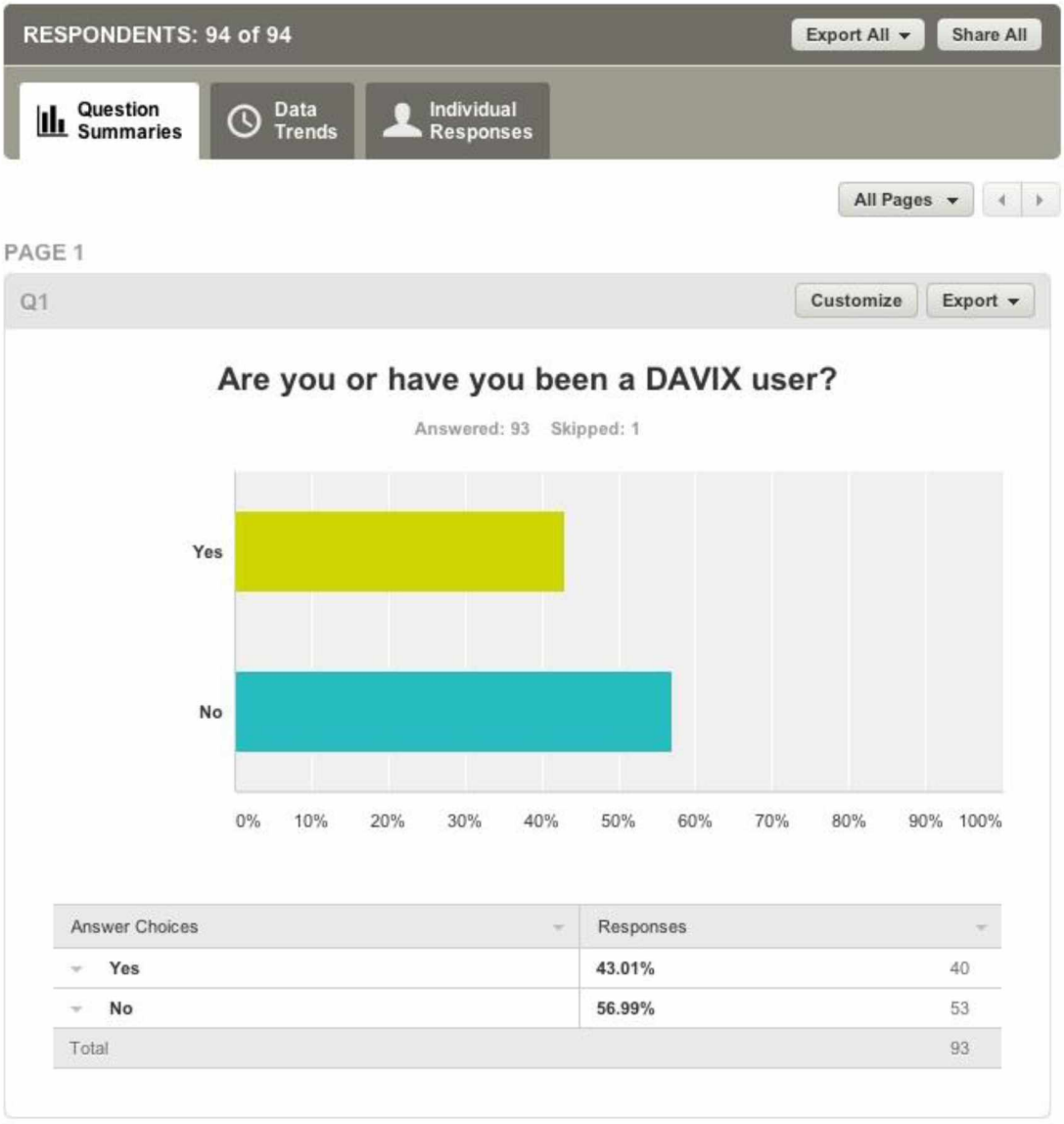
8. If we made it easy for you, would you want a Web service environment to use Web visualization libraries on DAVIX?
    a. Yes
    b. No
9. What are your biggest challenges when analyzing and visualizing security data?
    - getting the data
    - cleansing the data
    - parsing and normalizing the data

- understanding the data
- knowing what data to use for a specific problem
- storing and processing the data for later analysis
- analyzing the data, finding interesting/relevant areas in the data
- getting the data into the right format for analysis and visualization
- visualizing the data
- interpreting the visualizations
- sharing the analysis results
- Other (please specify)

10. Any other tools you need or want to see included in DAVIX? Is there any other input you would like to give us for the upcoming DAVIX release? Speak up now!

Appendix B - Survey Monkey Results

Survey Monkey results for the "DAVIX Release Survey" as of March 29, 2014.

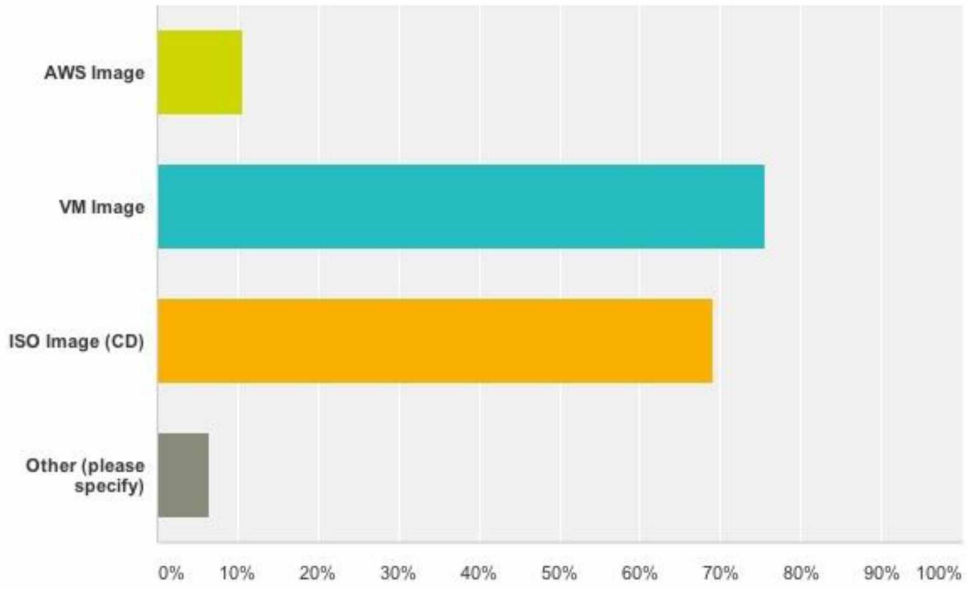


Answer Choices	Responses	
Yes	43.01%	40
No	56.99%	53
Total		93

Q2 Customize Export

How would you like to see DAVIX delivered?

Answered: 94 Skipped: 0



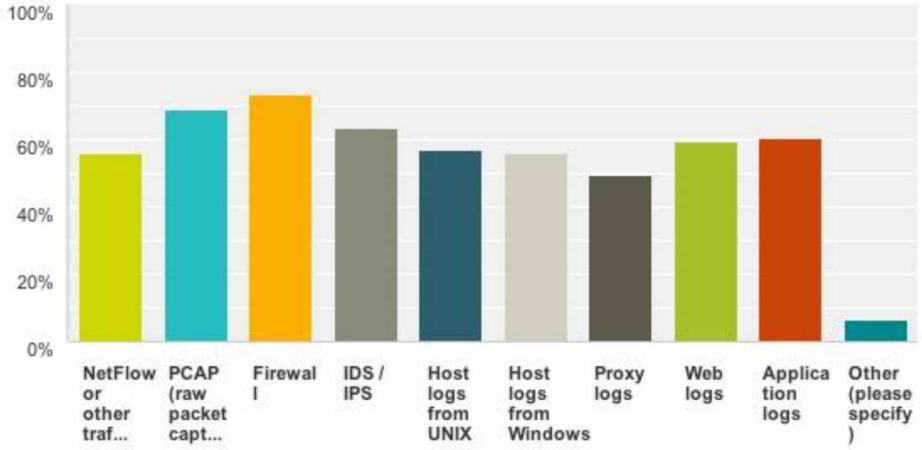
Answer Choices		Responses	
AWS Image		10.64%	10
VM Image		75.53%	71
ISO Image (CD)		69.15%	65
Other (please specify)	Responses	6.38%	6
Total Respondents: 94			

PAGE 2

Q3 Customize Export

What kind of data do you mainly work with?

Answered: 91 Skipped: 3

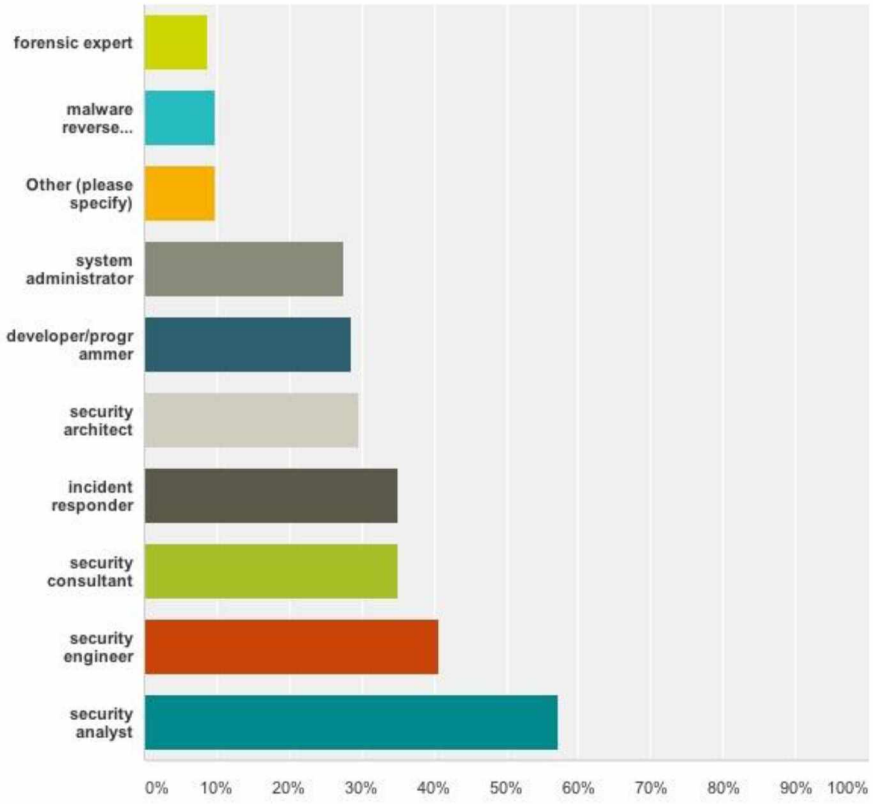


Answer Choices		Responses	
NetFlow or other traffic flows		56.04%	51
PCAP (raw packet captures)		69.23%	63
Firewall		73.63%	67
IDS / IPS		63.74%	58
Host logs from UNIX		57.14%	52
Host logs from Windows		56.04%	51
Proxy logs		49.45%	45
Web logs		59.34%	54
Application logs		60.44%	55
Other (please specify)	Responses	6.59%	6
Total Respondents: 91			

Q4 Customize Export

Are you a ...

Answered: 91 Skipped: 3



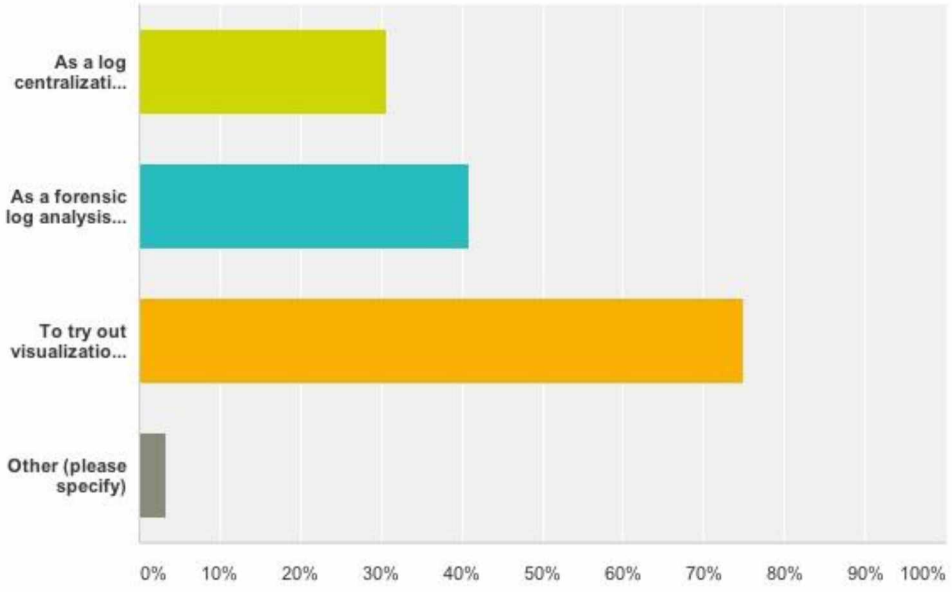
Answer Choices	Responses	
forensic expert	8.79%	8
malware reverse engineer	9.89%	9
Other (please specify) Responses	9.89%	9
system administrator	27.47%	25
developer/programmer	28.57%	26
security architect	29.67%	27
incident responder	35.16%	32
security consultant	35.16%	32
security engineer	40.66%	37
security analyst	57.14%	52
Total Respondents: 91		

PAGE 3

Q5 Customize Export

How do you intend to use or are already using DAVIX?

Answered: 88 Skipped: 6

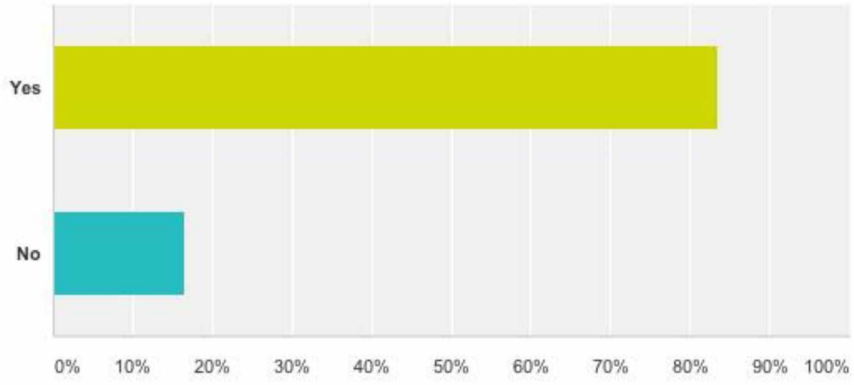


Answer Choices	Responses
As a log centralization and analysis workstation (using rsyslog, syslog-ng, logstash, or other to collect data and then analyze it)	30.68% 27
As a forensic log analysis distro (we get logs, load them on DAVIX and analyze them there)	40.91% 36
To try out visualization tools without having to install them myself	75.00% 66
Other (please specify) Responses	3.41% 3
Total Respondents: 88	

Q6 Customize Export

Are you fluent in UNIX?

Answered: 85 Skipped: 9



Answer Choices	Responses	
Yes	83.53%	71
No	16.47%	14
Total		85

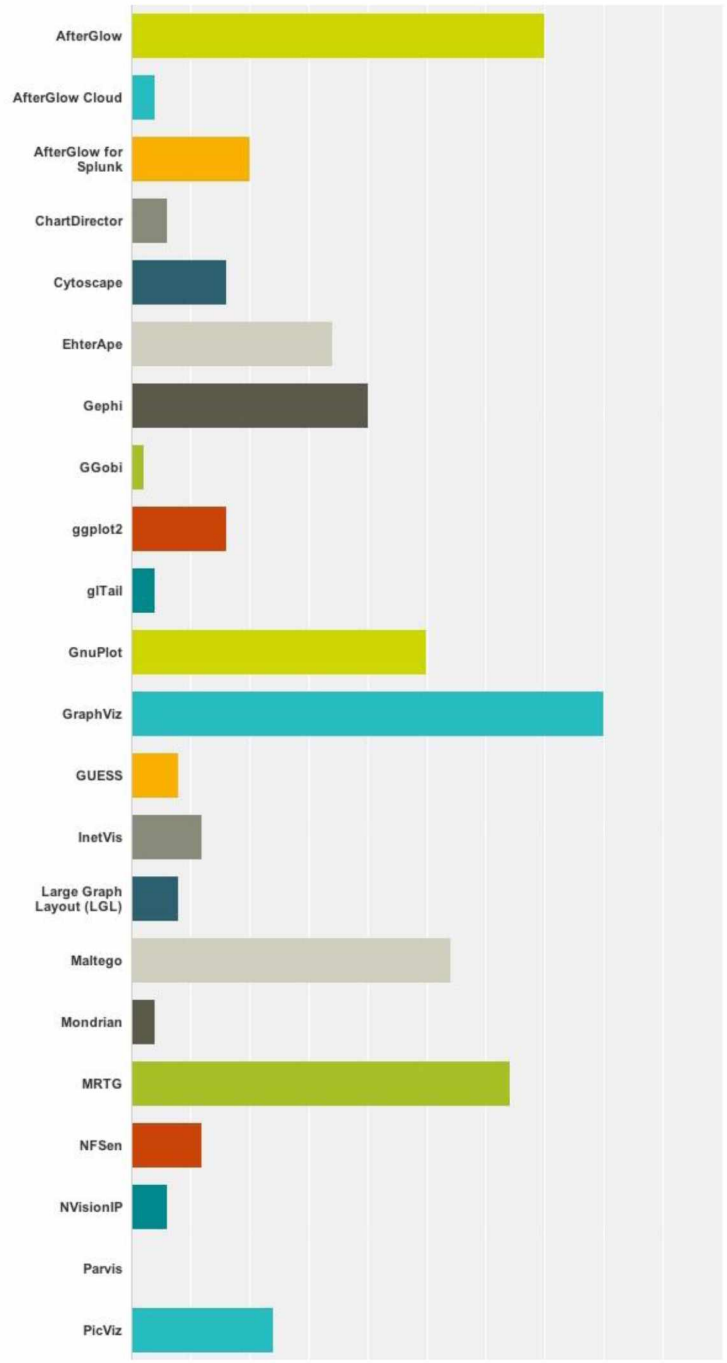
PAGE 4

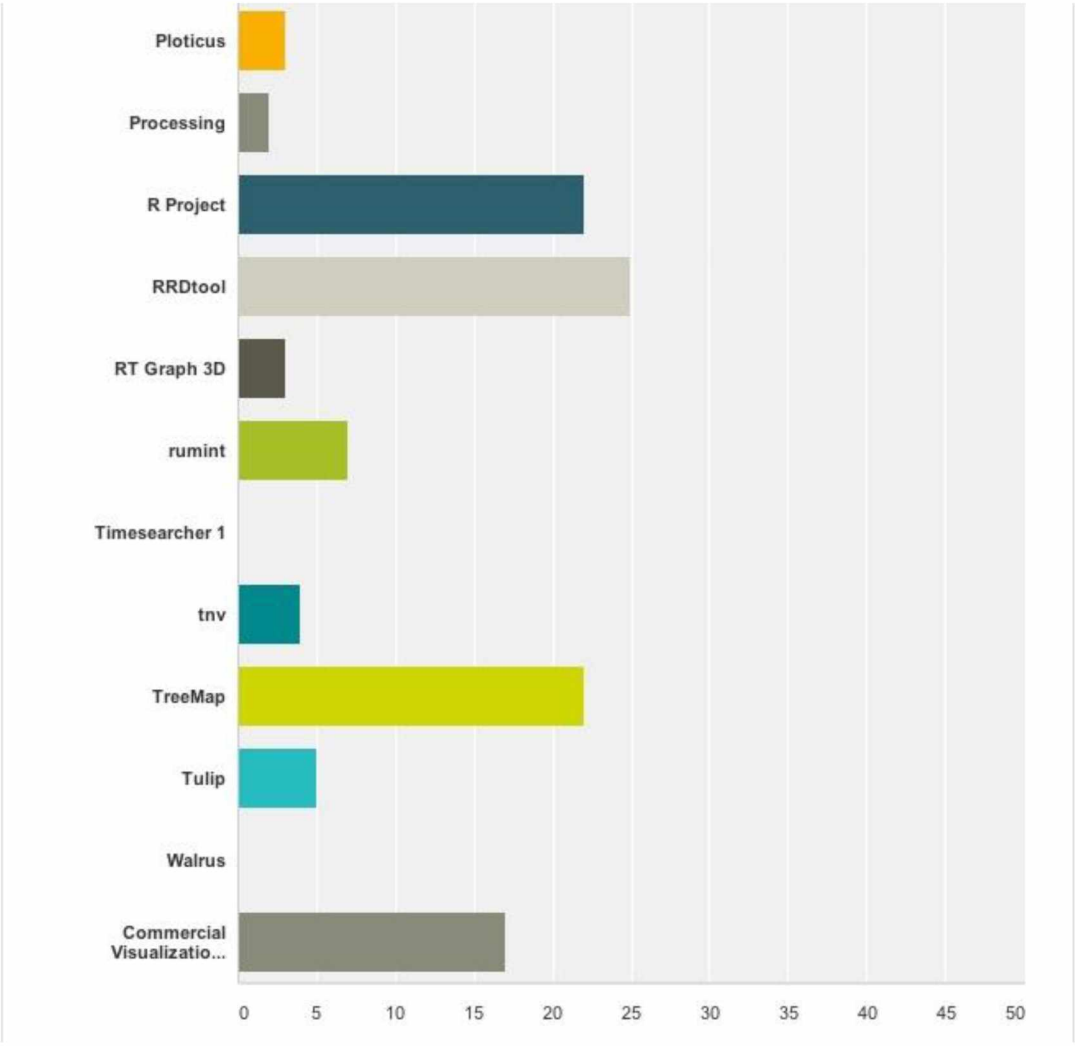
Q7

Customize Export

What (security) visualization tools are you using?

Answered: 74 Skipped: 20




Ploticus
Processing
R Project
RRDtool
RT Graph 3D
rumint
Timesearcher 1
tnv
TreeMap
Tulip
Walrus
Commercial Visualizatio...
0
5
10
15
20
25
30
35
40
45
50

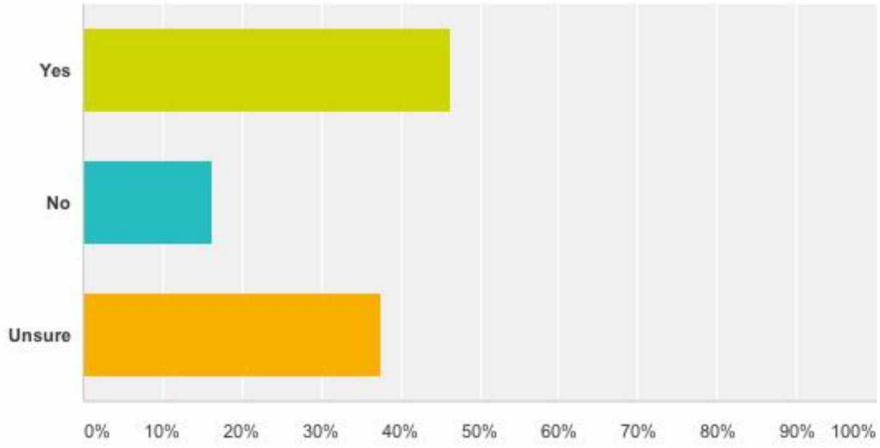
Answer Choices		Responses	
AfterGlow		47.30%	35
AfterGlow Cloud		2.70%	2
AfterGlow for Splunk		13.51%	10
ChartDirector		4.05%	3
Cytoscape		10.81%	8
EhterApe		22.97%	17
Gephi		27.03%	20
GGobi		1.35%	1
ggplot2		10.81%	8
glTail		2.70%	2
GnuPlot		33.78%	25
GraphViz		54.05%	40
GUESS		5.41%	4
InetVis		8.11%	6
Large Graph Layout (LGL)		5.41%	4
Maltego		36.49%	27
Mondrian		2.70%	2
MRTG		43.24%	32
NFSen		8.11%	6
NVisionIP		4.05%	3
Parvis		0.00%	0
PicViz		16.22%	12
Ploticus		4.05%	3
Processing		2.70%	2
R Project		29.73%	22
RRDtool		33.78%	25
RT Graph 3D		4.05%	3
rumint		9.46%	7
Timesearcher 1		0.00%	0
tnv		5.41%	4
TreeMap		29.73%	22
Tulip		6.76%	5
Walrus		0.00%	0
Commercial Visualization Tool or Other (please specify)	Responses	22.97%	17
Total Respondents: 74			

PAGE 5

Q8 Customize Export

If we made it easy for you, would you want a Web service environment to use Web visualization libraries on DAVIX?

Answered: 80 Skipped: 14

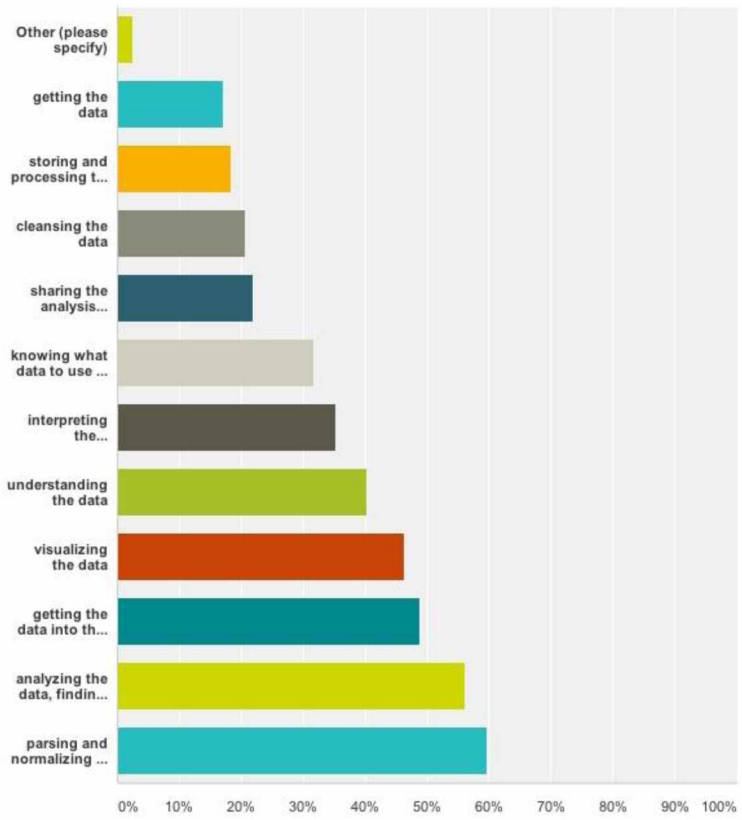


Answer Choices	Responses	
Yes	46.25%	37
No	16.25%	13
Unsure	37.50%	30
Total		80

Q9 Customize Export

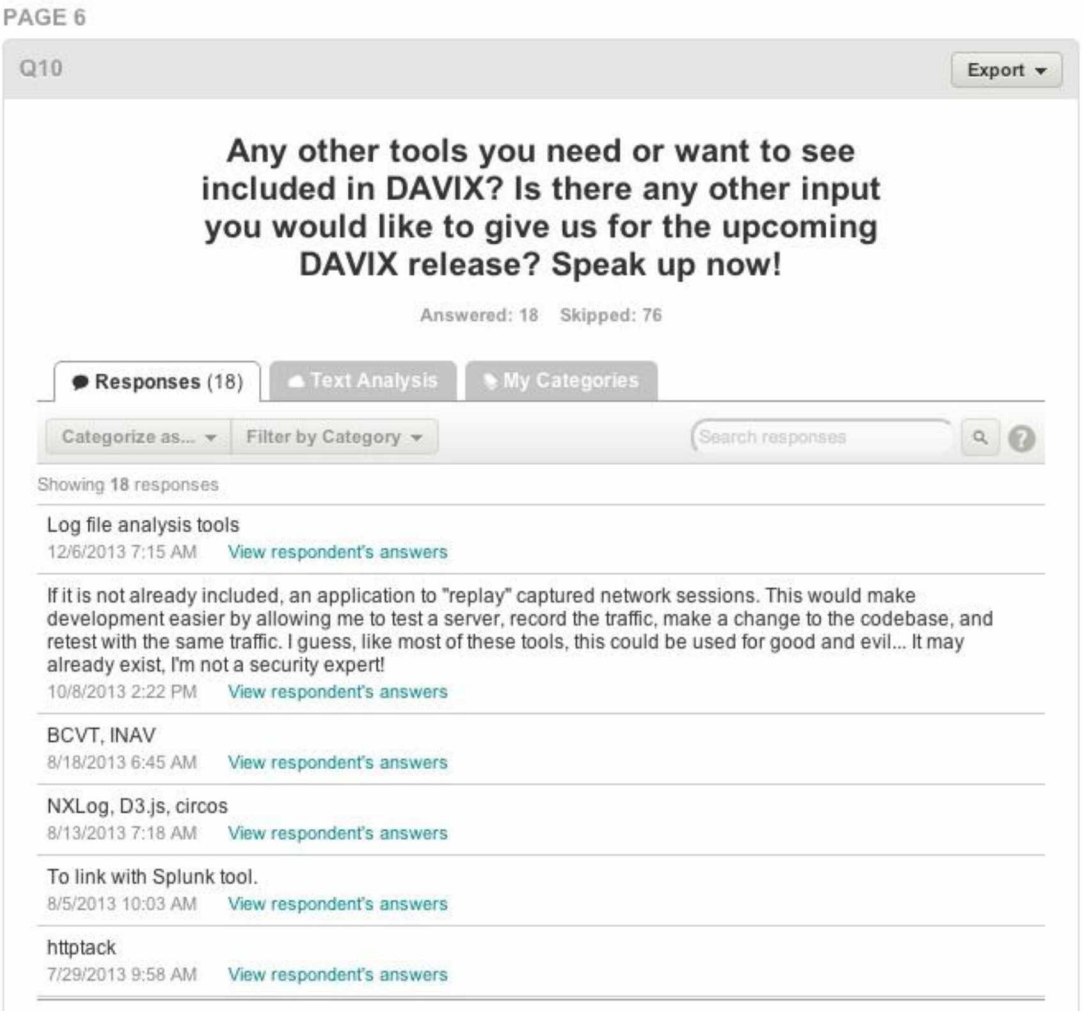
What are your biggest challenges when analyzing and visualizing security data?

Answered: 82 Skipped: 12



Answer Choices	Responses	
Other (please specify) Responses	2.44%	2
getting the data	17.07%	14
storing and processing the data for later analysis	18.29%	15
cleansing the data	20.73%	17
sharing the analysis results	21.95%	18
knowing what data to use for a specific problem	31.71%	26
interpreting the visualizations	35.37%	29
understanding the data	40.24%	33
visualizing the data	46.34%	38
getting the data into the right format for analysis and visualization	48.78%	40
analyzing the data, finding interesting/relevant areas in the data	56.10%	46
parsing and normalizing the data	59.76%	49

Total Respondents: 82



The results in the view for this question were truncated so a complete list follows:

- Log file analysis tools
- If it is not already included, an application to "replay" captured network sessions. This would make development easier by allowing me to test a server,

record the traffic, make a change to the codebase, and retest with the same traffic. I guess, like most of these tools, this could be used for good and evil... It may already exist, I'm not a security expert!

- BCVT, INAV
- NXLog, D3.js, circos
- To link with Splunk tool.
- httptack
- bro, ELSA
- BRO, Elsa
- Bro
- none at this time
- Have a look at security onion. If you can make it a place we can install both on the same workstation
- Have you looked at Moloch at all? It is a fairly new tool for capturing/analyzing pcap captures https://github.com/aol/moloch
- None that I'm aware of.
- a sumologic connector

- I don't know if they fit, but I would like to see some of the javascript tools, like highcharts and d3, as part of a Web infrastructure to visualize data.
- Wireshark of course
- proxy tools : burp, nikto, owasp zap, ...
- https://github.com/stricaud/faup

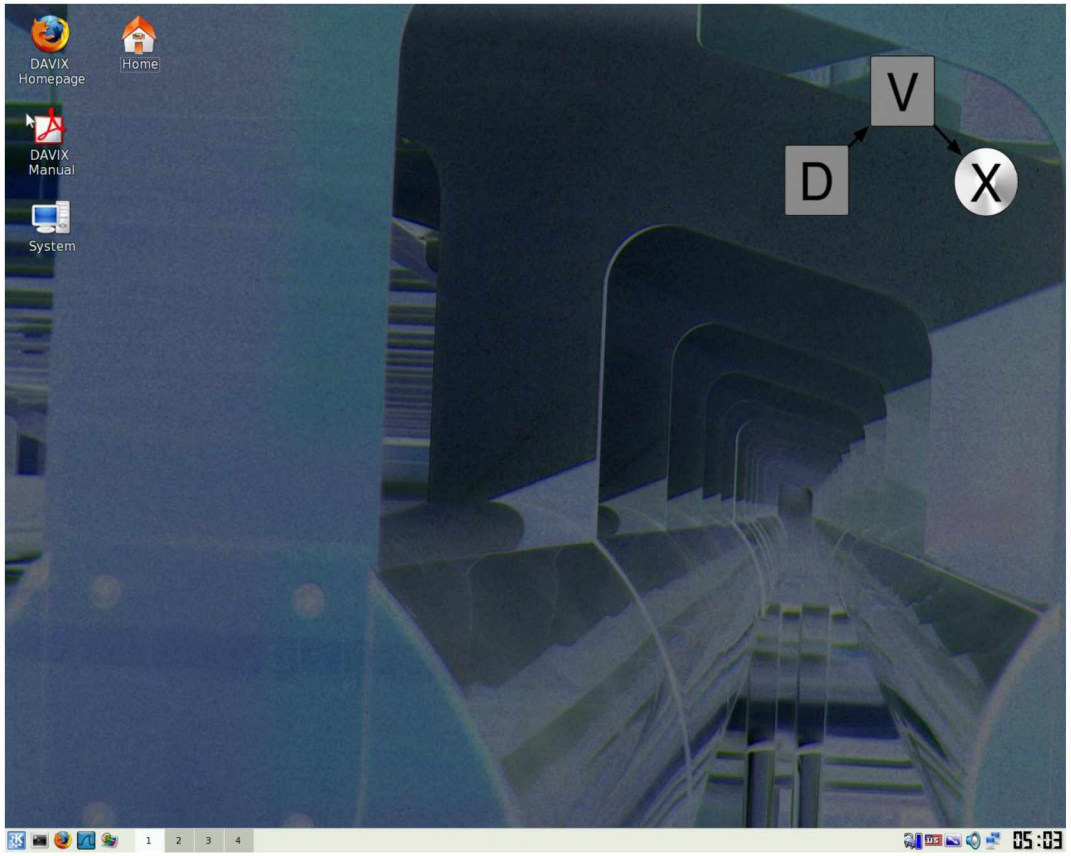
Appendix C - Tool Example – InetVis

InetVis allows real-time visualization of network

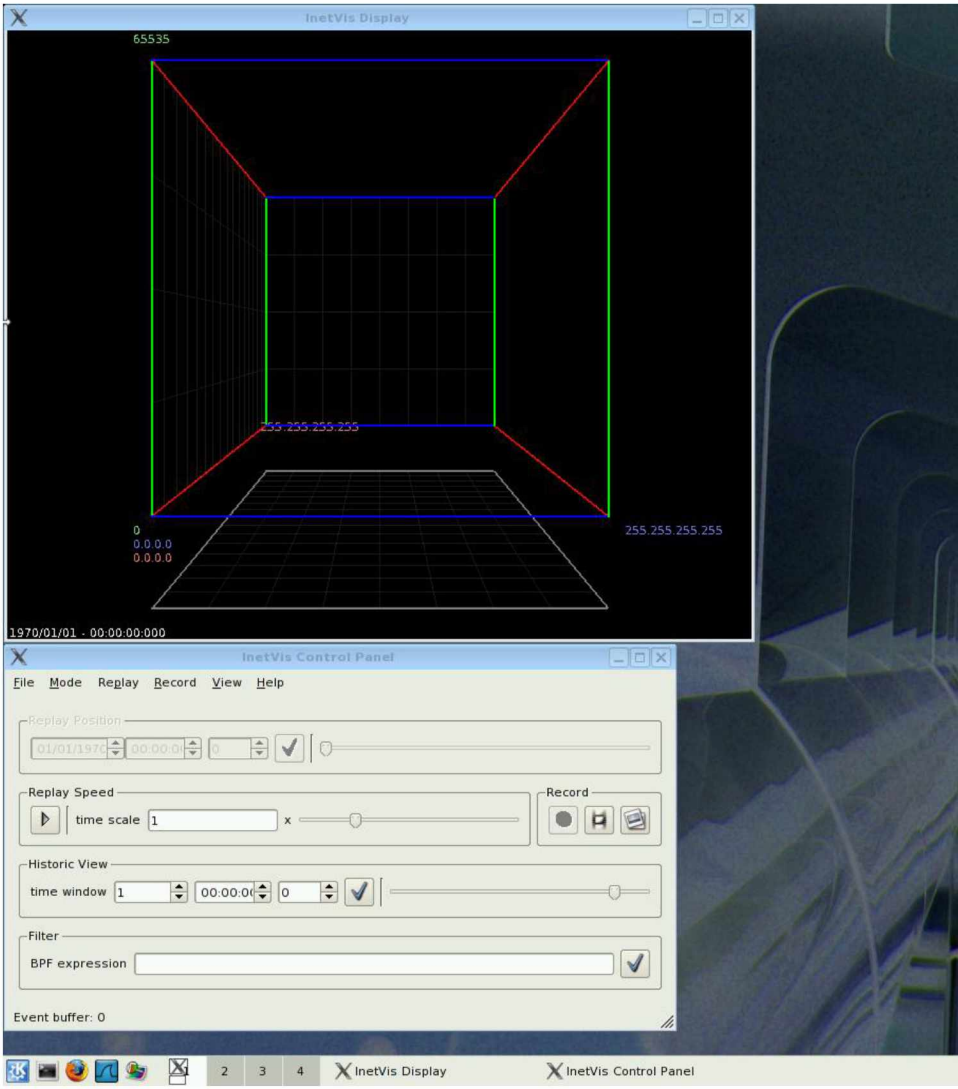
http://www.cs.ru.ac.za/research/g02v2468/inetvis.html

traffic as a scatter plot and is available at

I ran InetVis from DAVIX and thus began at the DAVIX home screen:



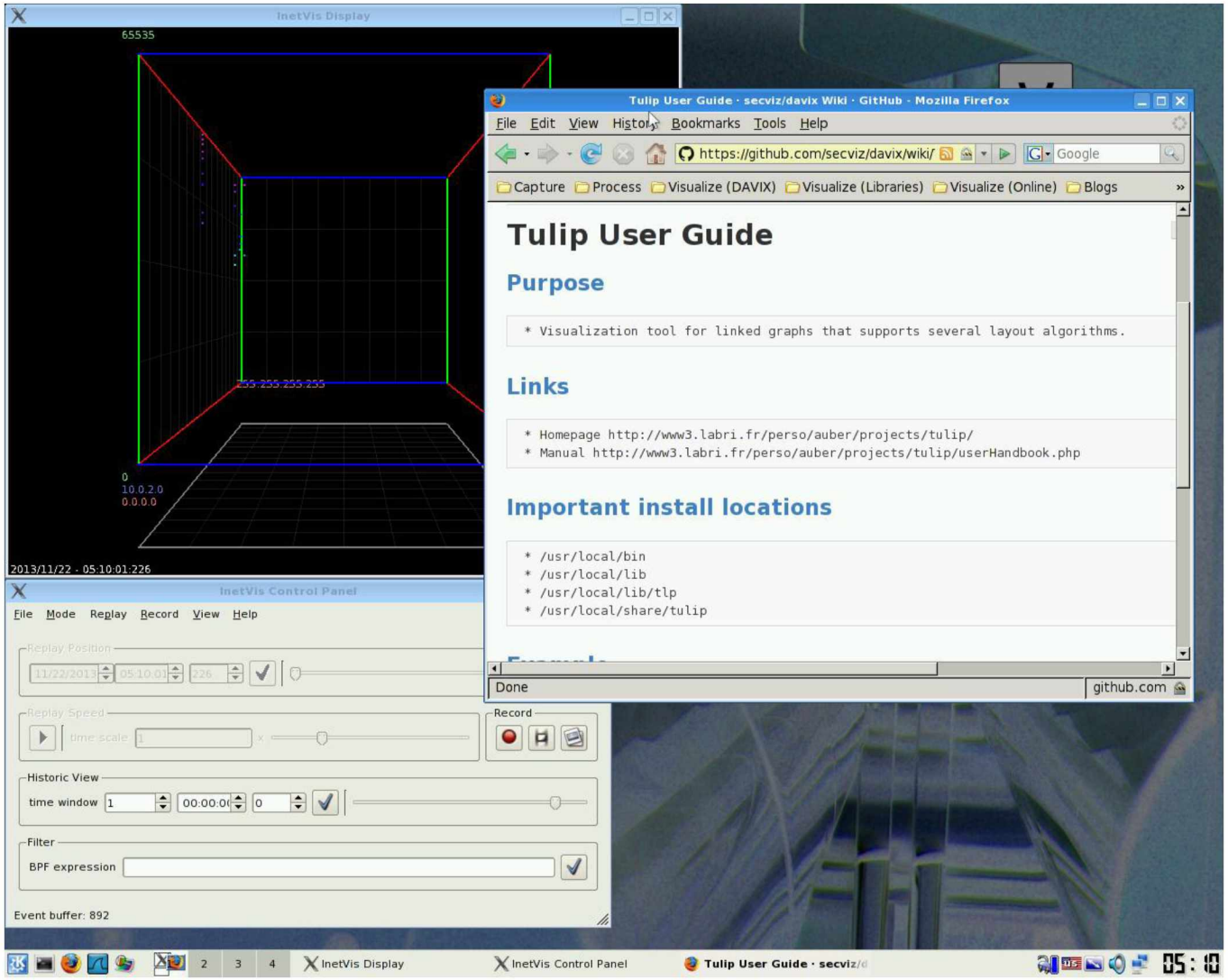
Then I opened InetVis.



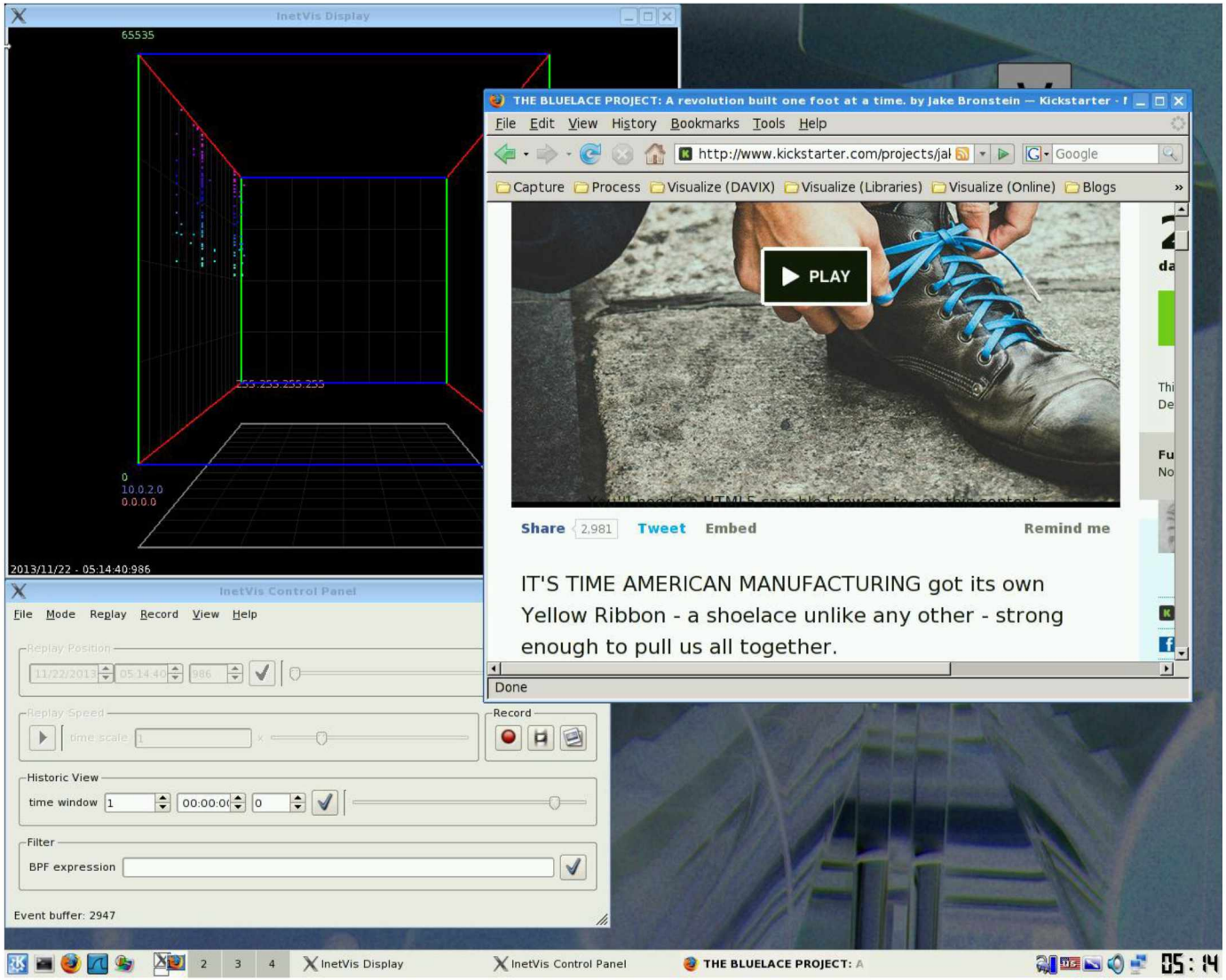

V
D
X
05:05

I went to the Control Panel -> Mode Menu and set “Monitor Local Host”. The timestamp began running.

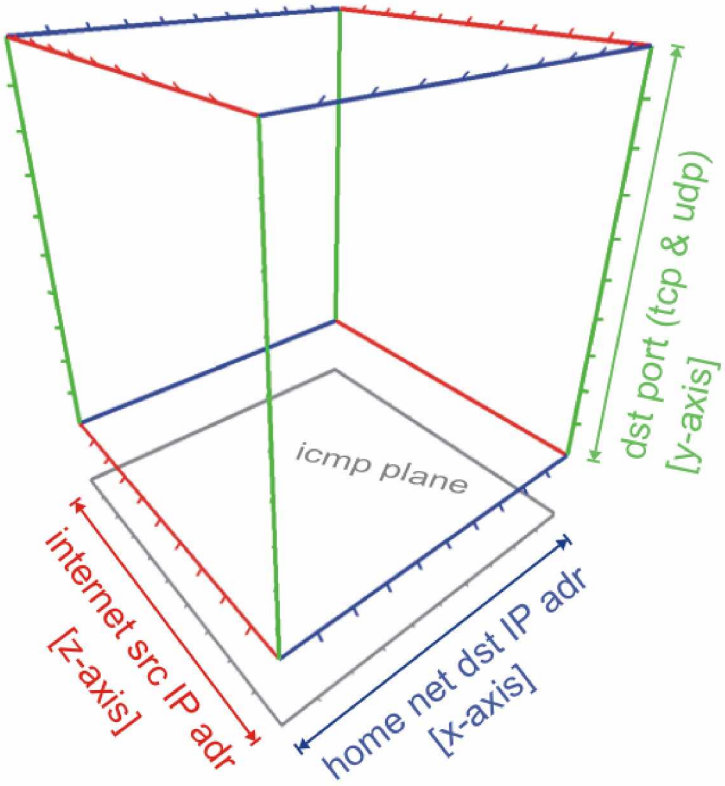
I then began browsing the Internet. You can already see the scatter plot beginning to take shape.



The more browsing I did the more points were plotted as various Internet source addresses and ports were accessed.



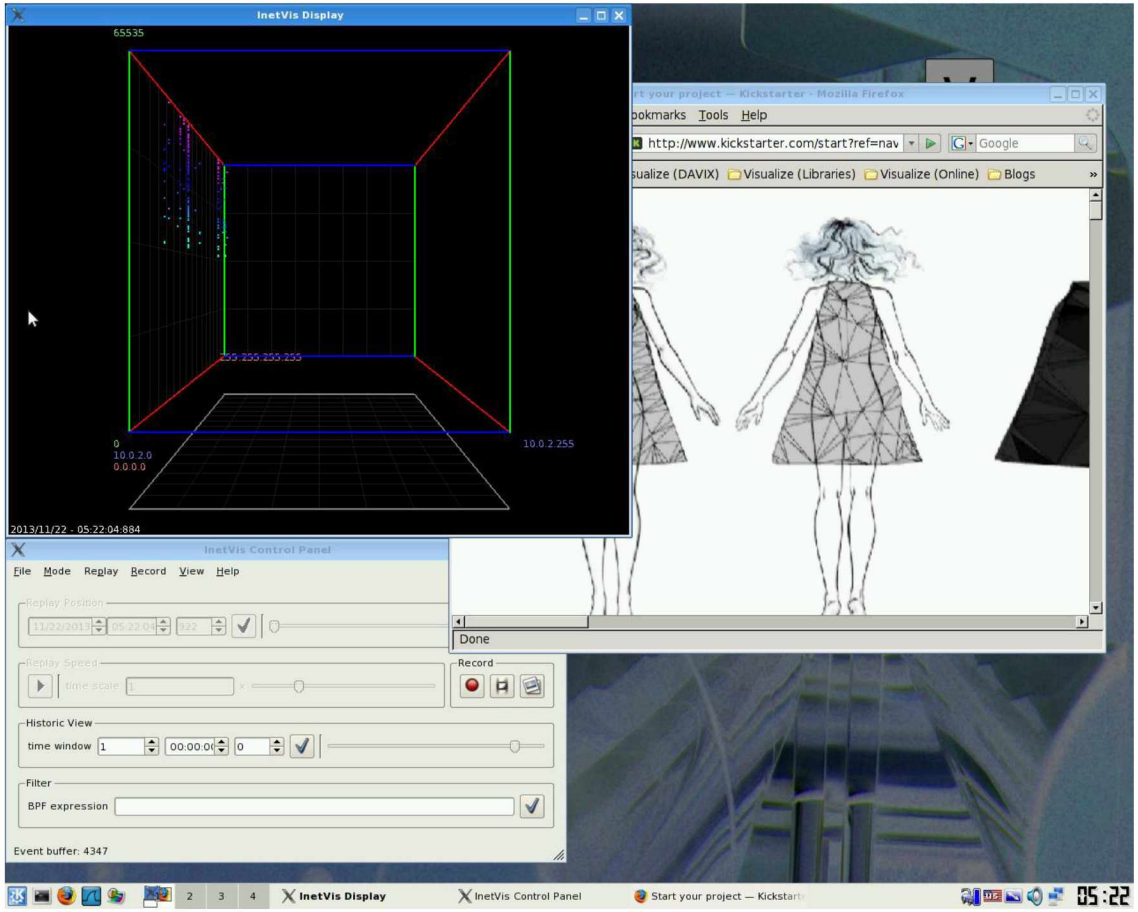
The InetVis scatter plot cube is diagrammed on their site as follows:



InetVis Diagram – Source: http://www.cs.ru.ac.za/research/g02v2468/inetvis.html

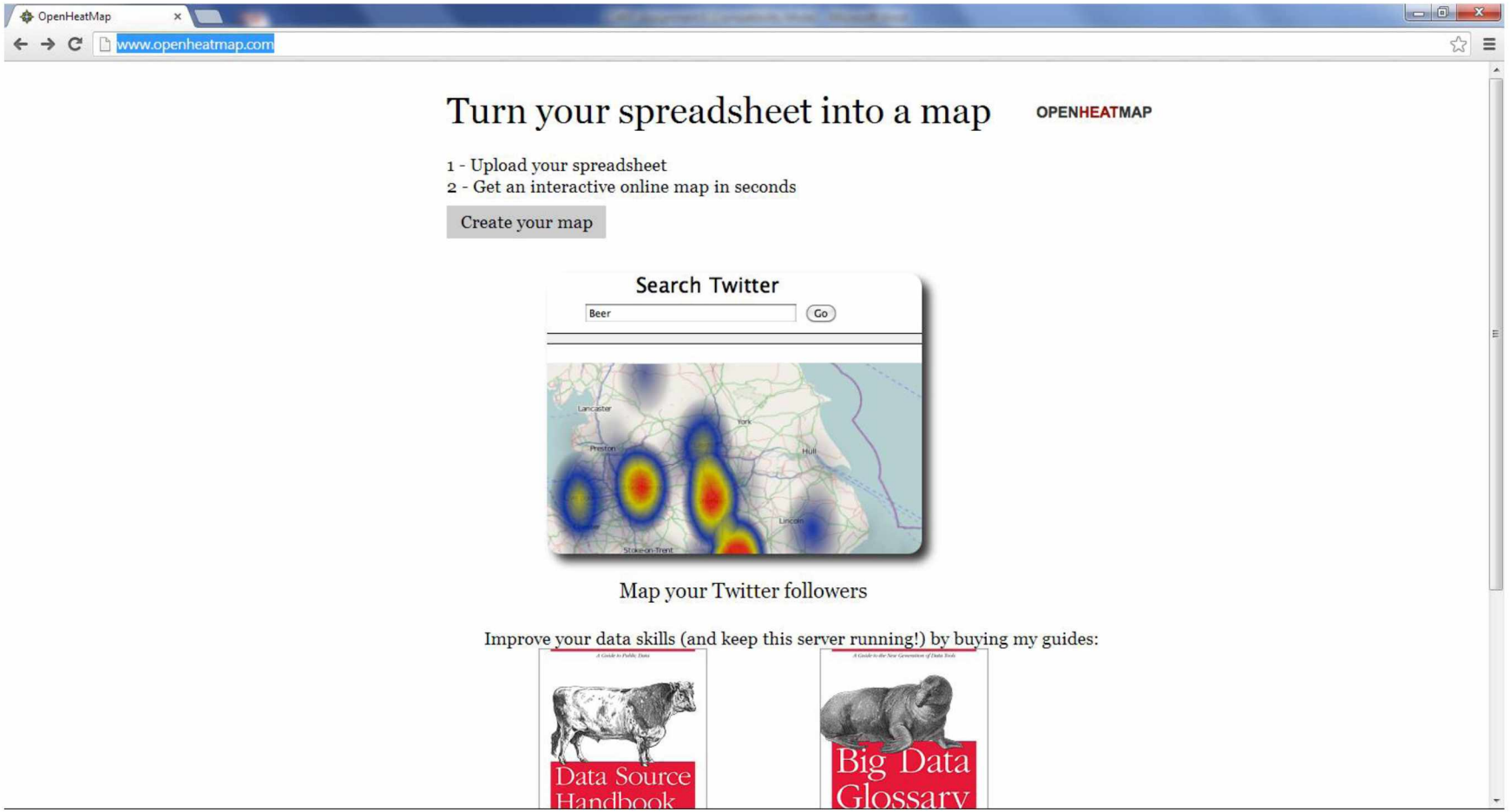
The home network is plotted along the blue x-axis, the source address along the red z-axis and the ports on the green y-axis.

As you can see by the scatter plot since I am browsing the web solely from the VM containing DAVIX there's really only relevant information on the red z-axis and green y-axis which shows us the source addresses I'm visiting and ports.



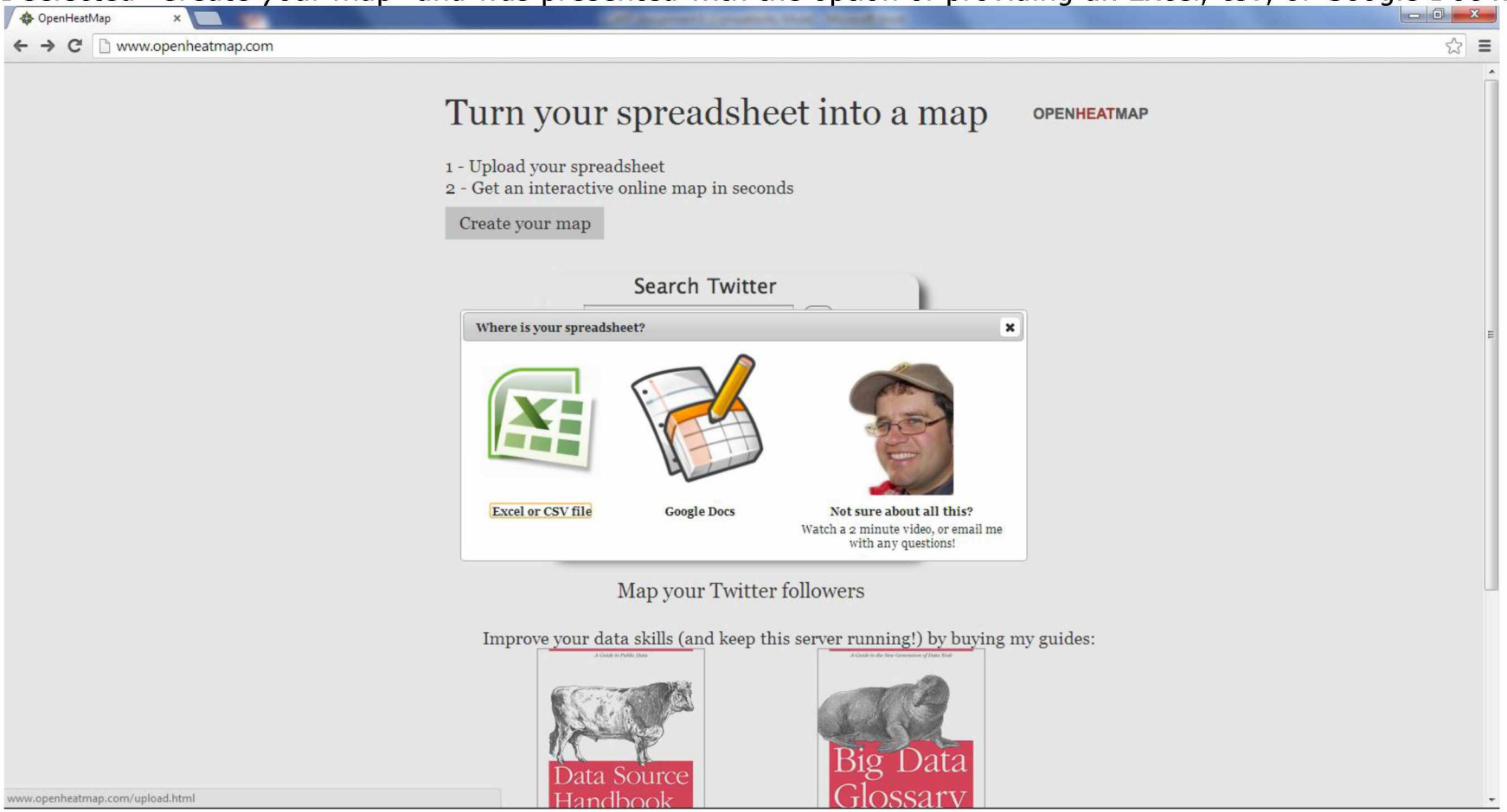
Appendix D - Tool Example – OpenHeatMap

OpenHeatMap provides a geographic heat map based on an Excel or csv file. It is a web application available at http://www.openheatmap.com/

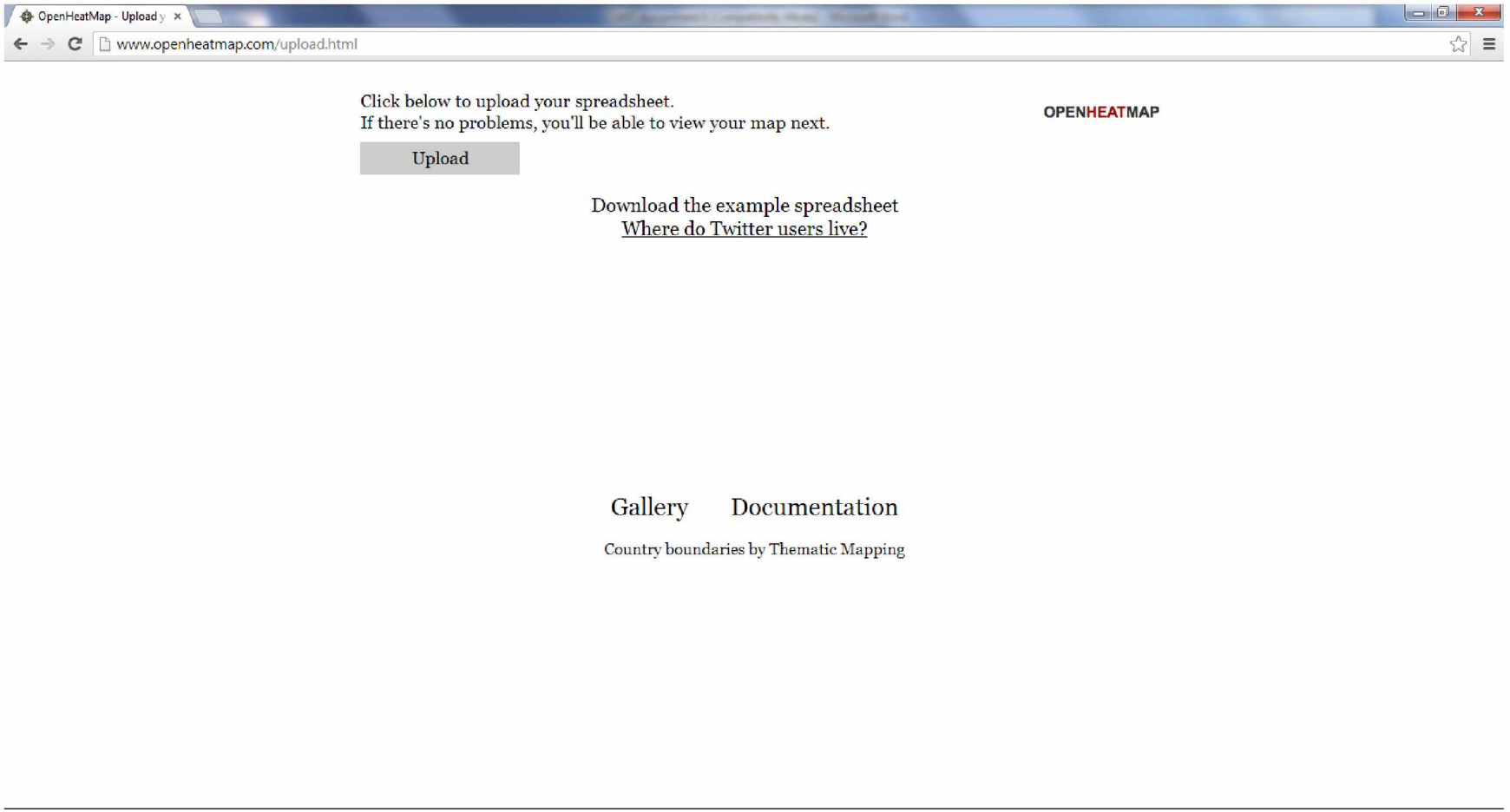


100

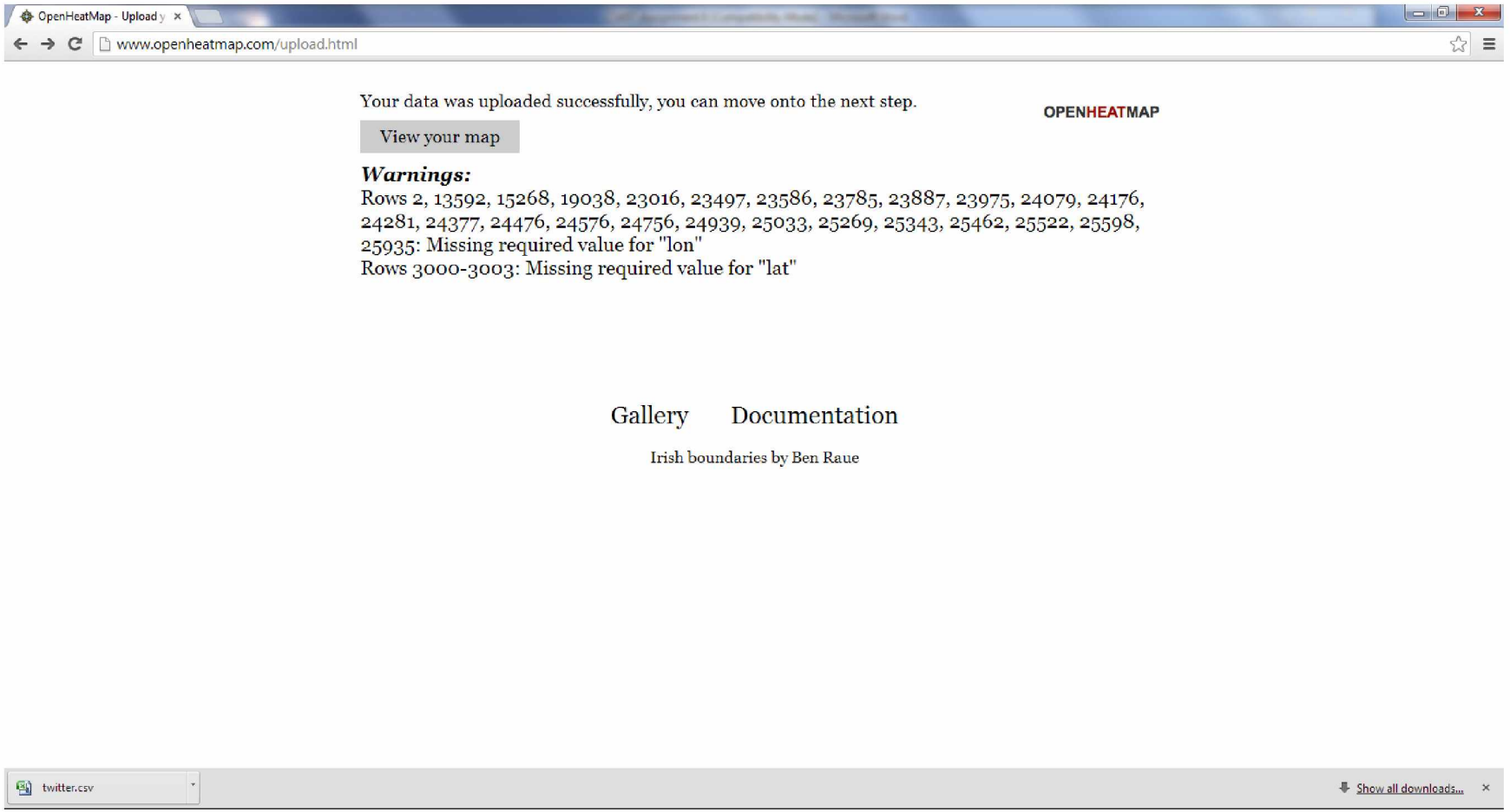
I selected "Create your map" and was presented with the option of providing an Excel, csv, or Google Doc file.



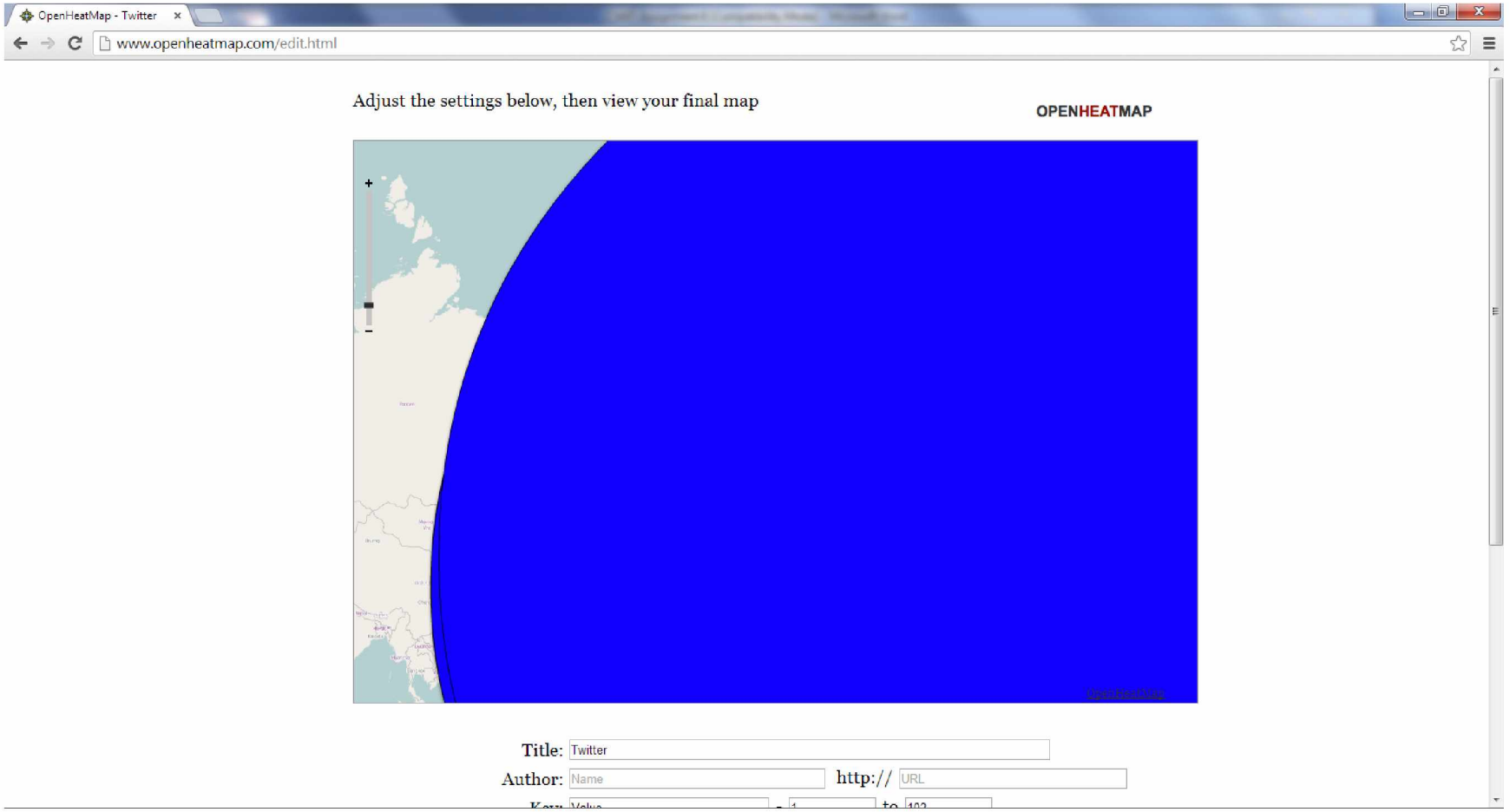
I selected Excel or CSV file. The site then scrolled through a few sample files that could be used; I selected "Where do Twitter users live" and downloaded the file.



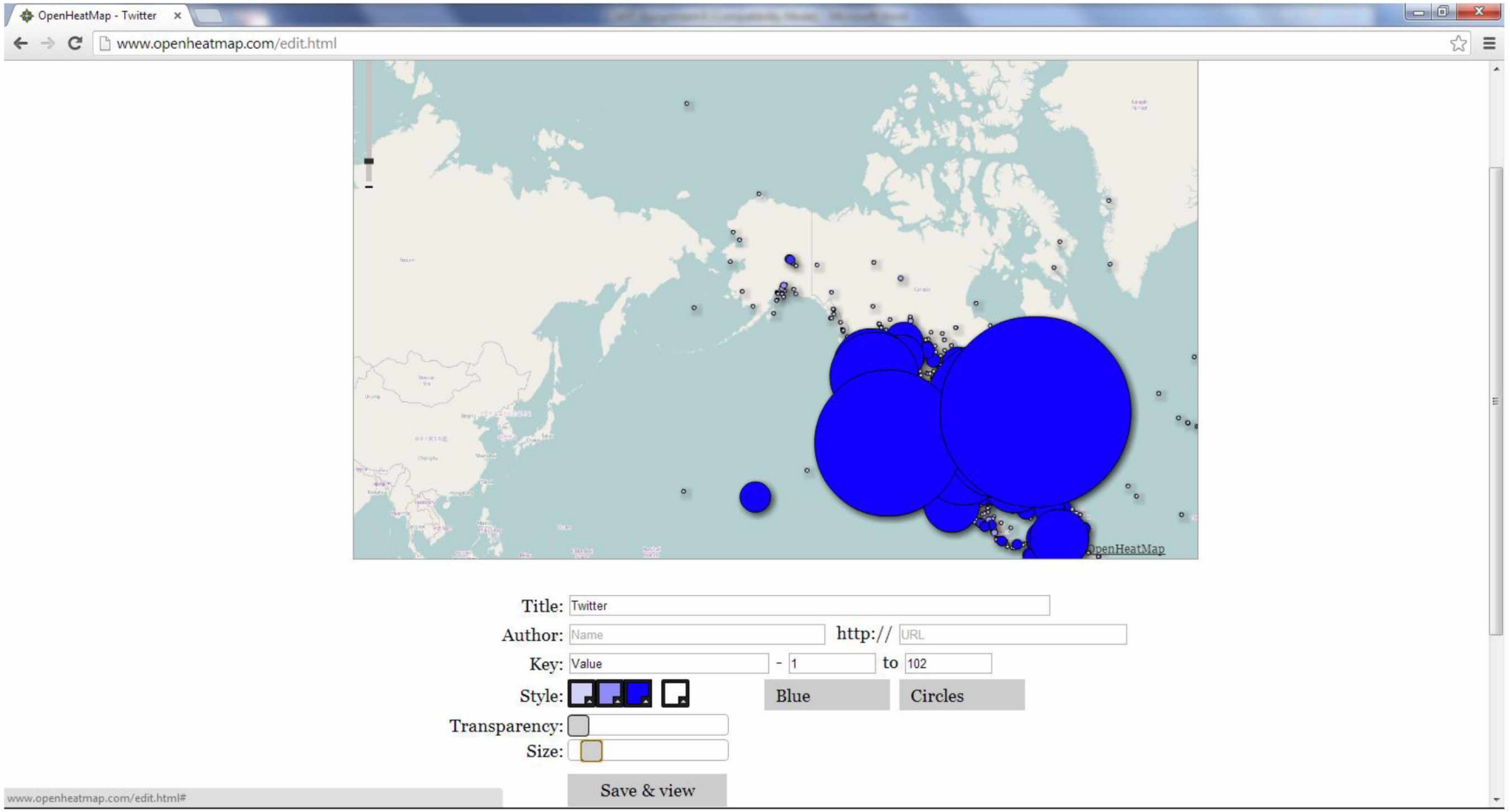
Then I clicked Upload and located the file I'd just downloaded. I received some warnings, which was interesting considering this was a sample file they provided.



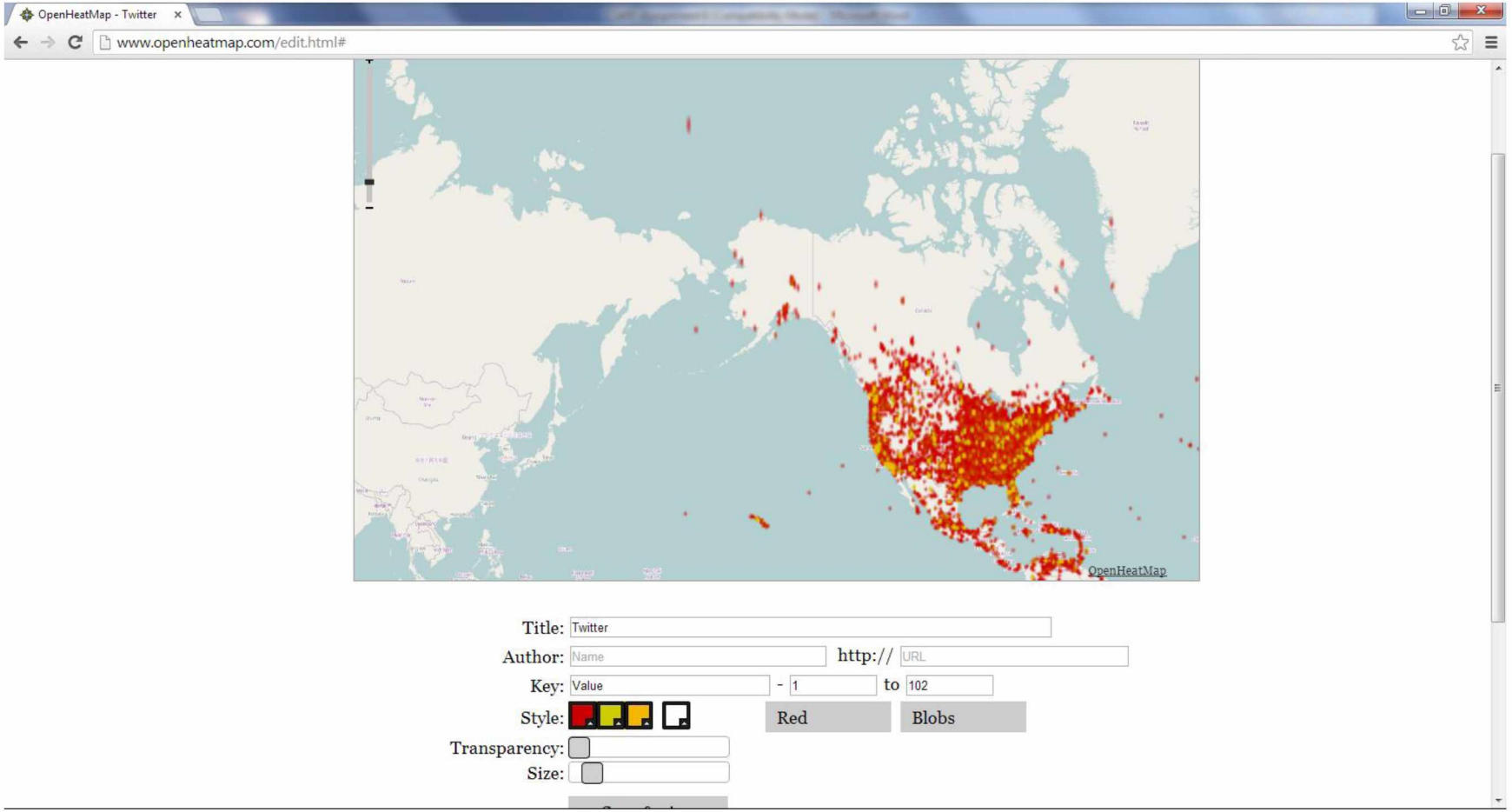
I clicked "View your map". The initial view was almost unreadable.



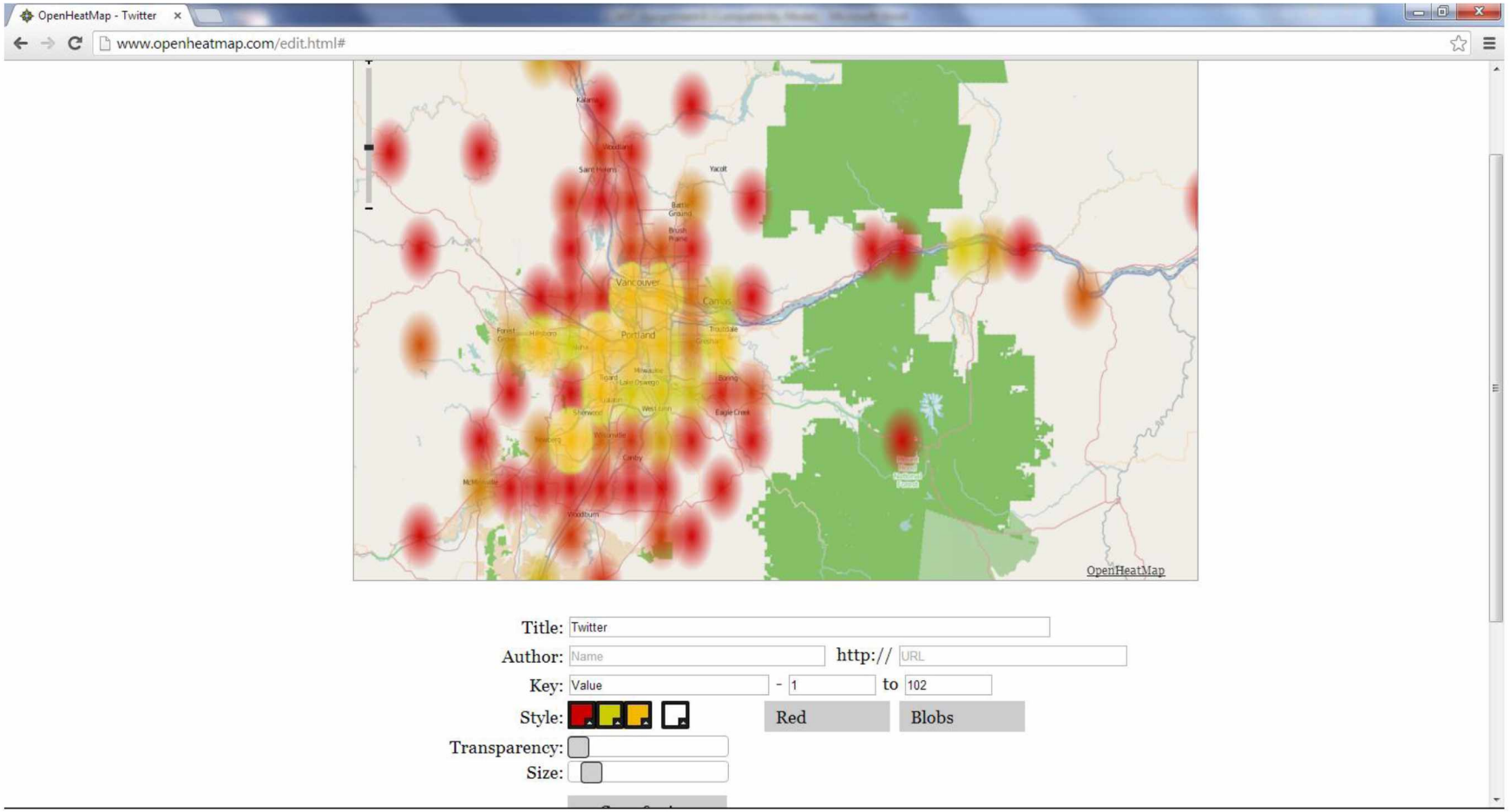
I changed the Size to a much smaller value.



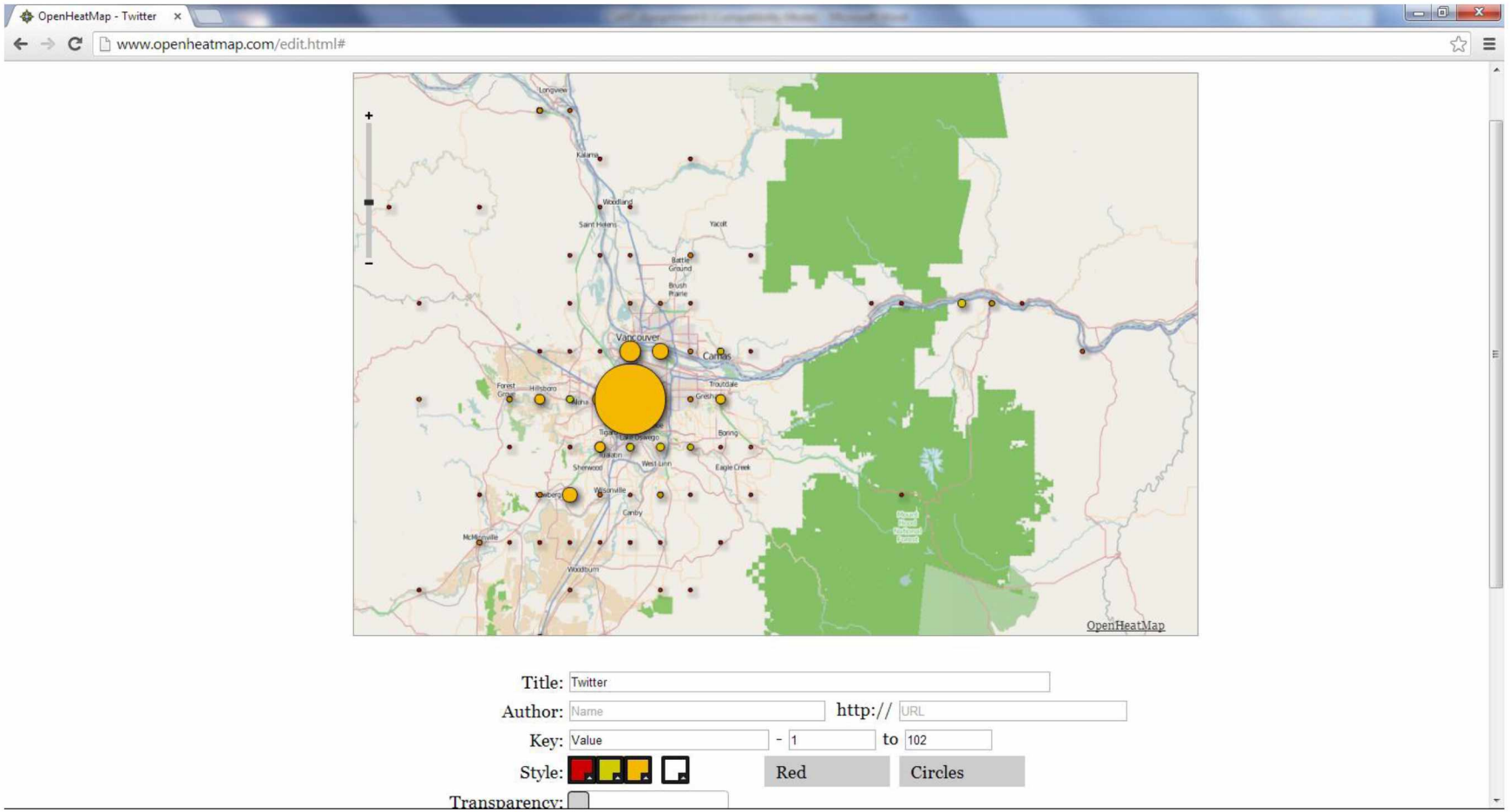
I changed the color scheme to Red and set it to display Blobs. The areas are now color coded with orange being the highest density and red being lower density.



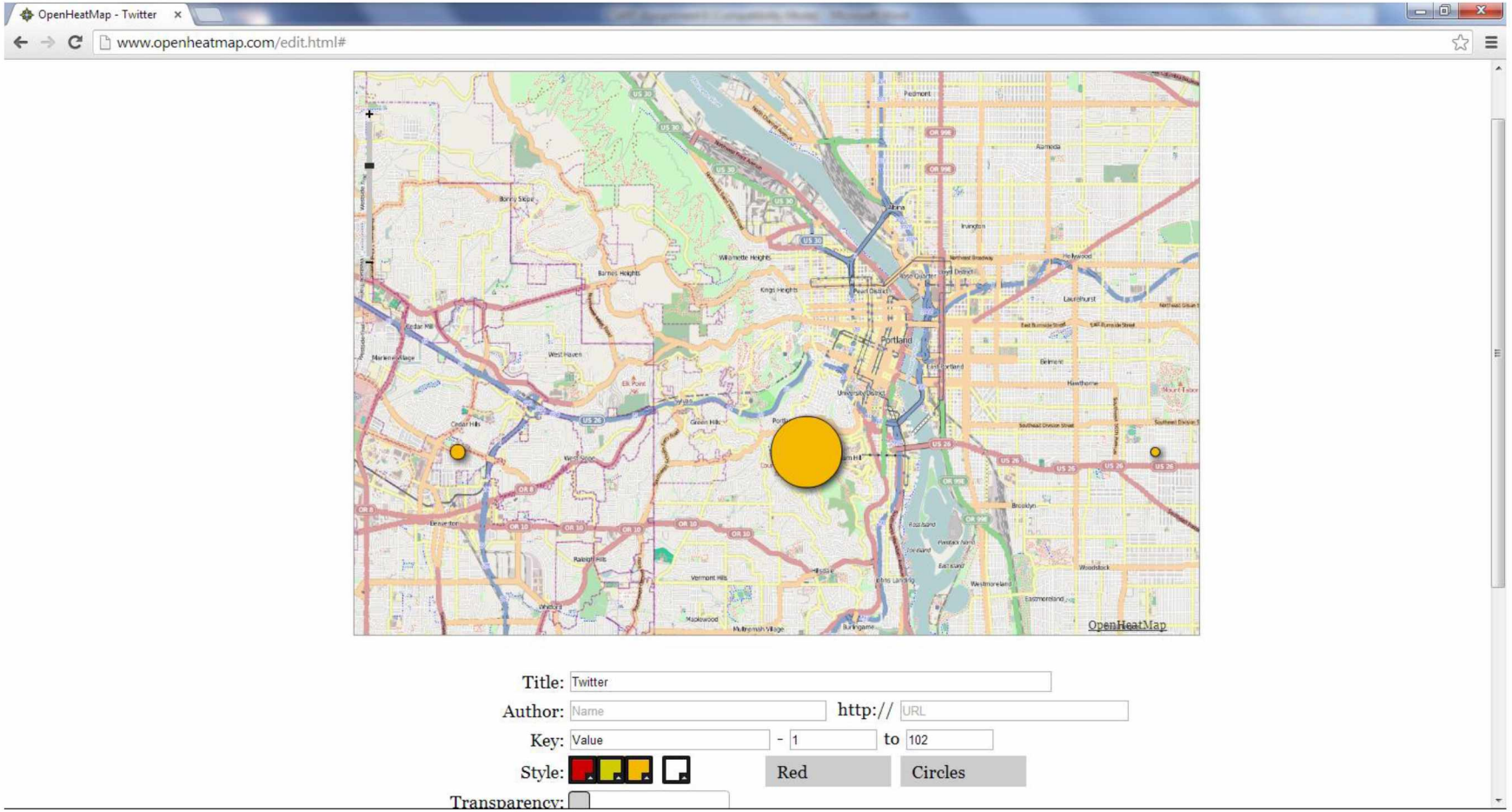
I zoomed in on the Portland area.



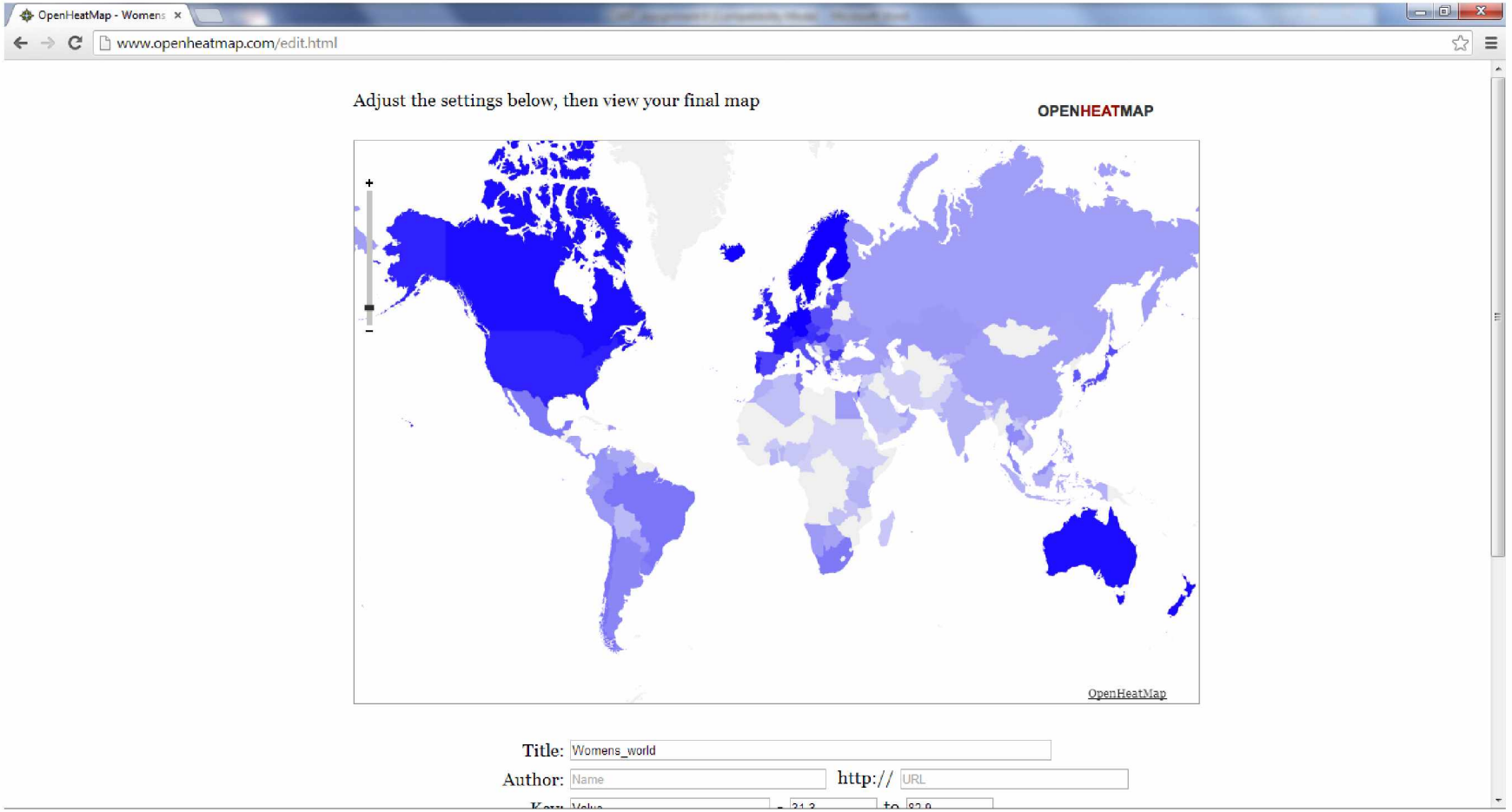
I changed it from Blobs to Circles. This was a little easier to view and definitely demonstrated that the Portland city center had the highest density of Twitter users.



I zoomed in further and here the data became a little useless as the latitude/longitude data did not provide sufficient granularity to precisely locate the Twitter users, just the city centers they were in.

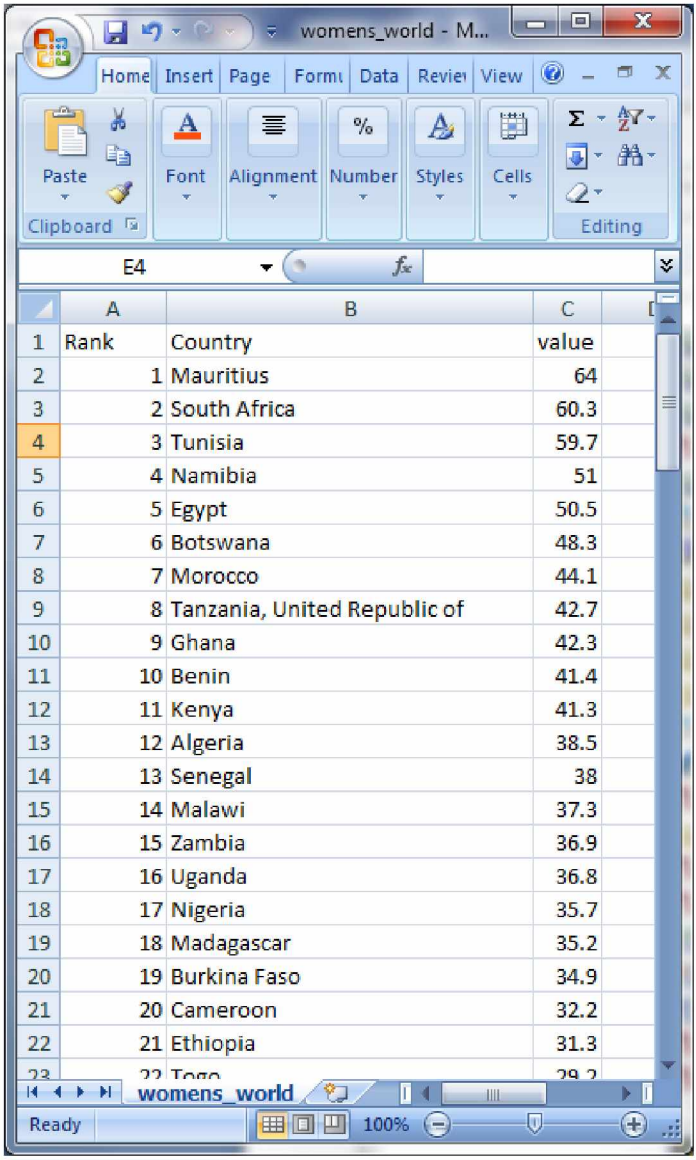


I went back to the main site and went through the creation process so I could download the sample file "The worst places in the world to be a woman." I uploaded it and viewed my map. Note how the granularity is now on the country level and not anything more specific.



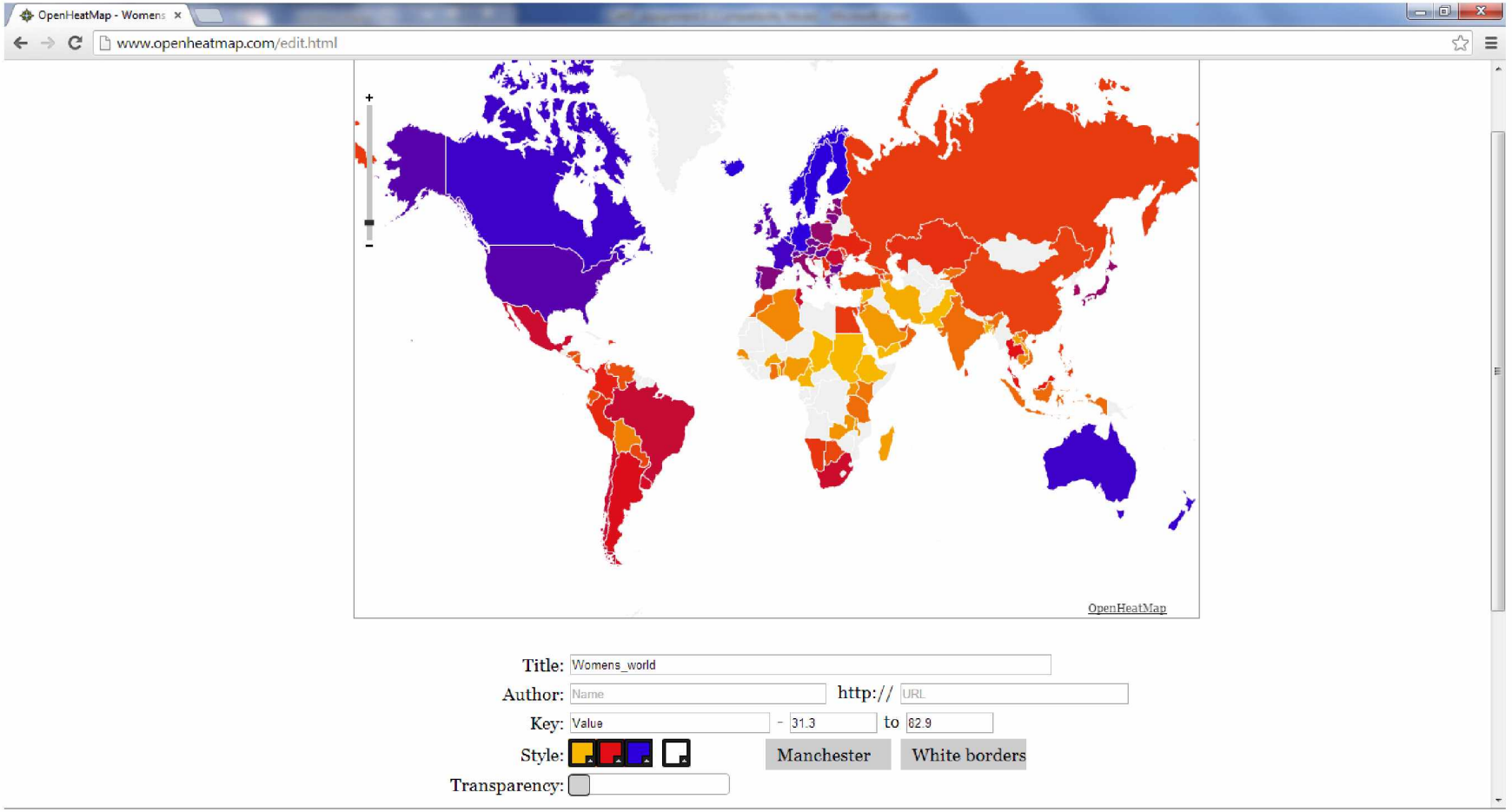
110

This is because the data file has specified location by Country.

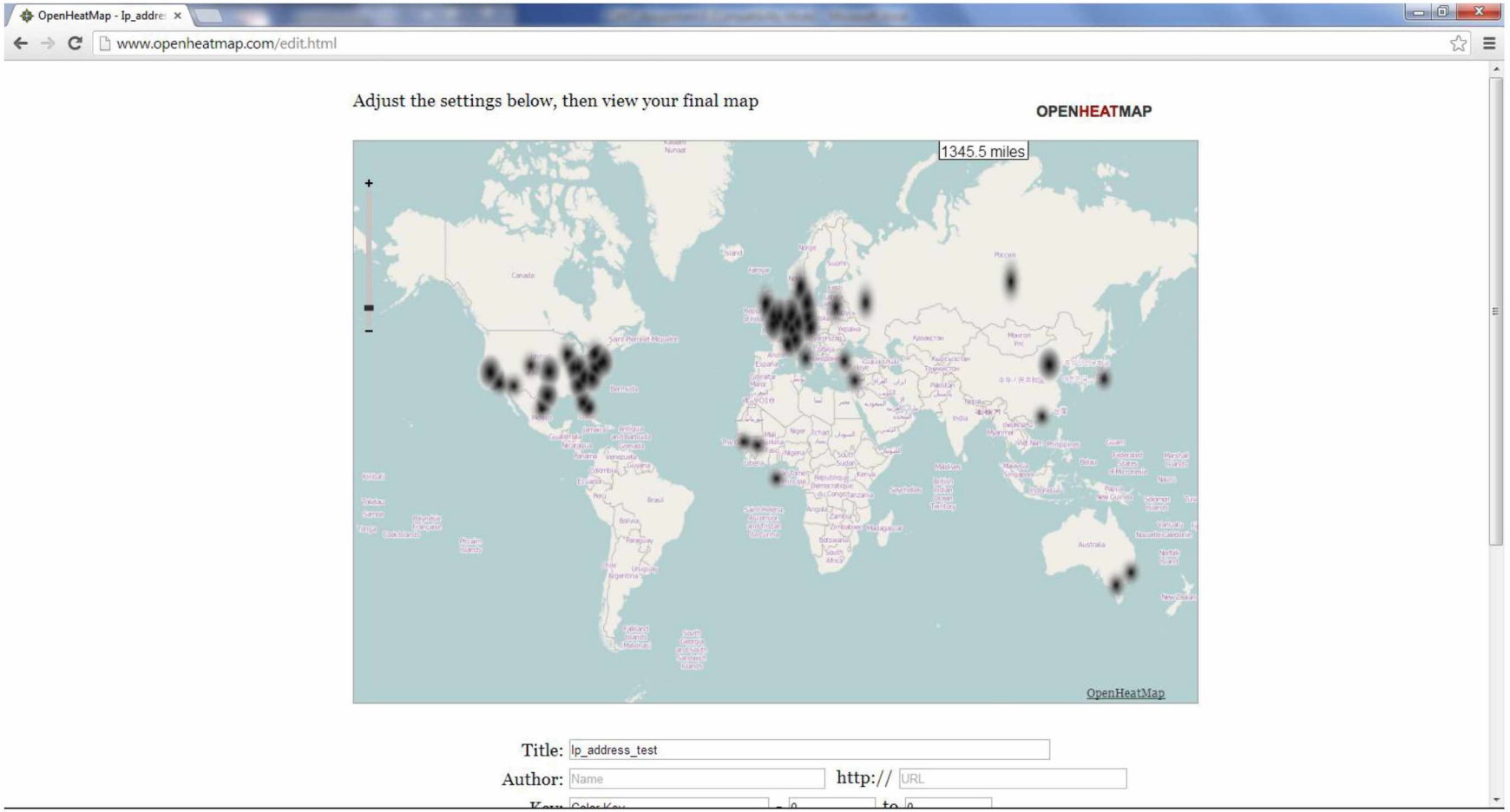


	A	B	C
1	Rank	Country	value
2	1	Mauritius	64
3	2	South Africa	60.3
4	3	Tunisia	59.7
5	4	Namibia	51
6	5	Egypt	50.5
7	6	Botswana	48.3
8	7	Morocco	44.1
9	8	Tanzania, United Republic of	42.7
10	9	Ghana	42.3
11	10	Benin	41.4
12	11	Kenya	41.3
13	12	Algeria	38.5
14	13	Senegal	38
15	14	Malawi	37.3
16	15	Zambia	36.9
17	16	Uganda	36.8
18	17	Nigeria	35.7
19	18	Madagascar	35.2
20	19	Burkina Faso	34.9
21	20	Cameroon	32.2
22	21	Ethiopia	31.3
23	22	Togo	29.2

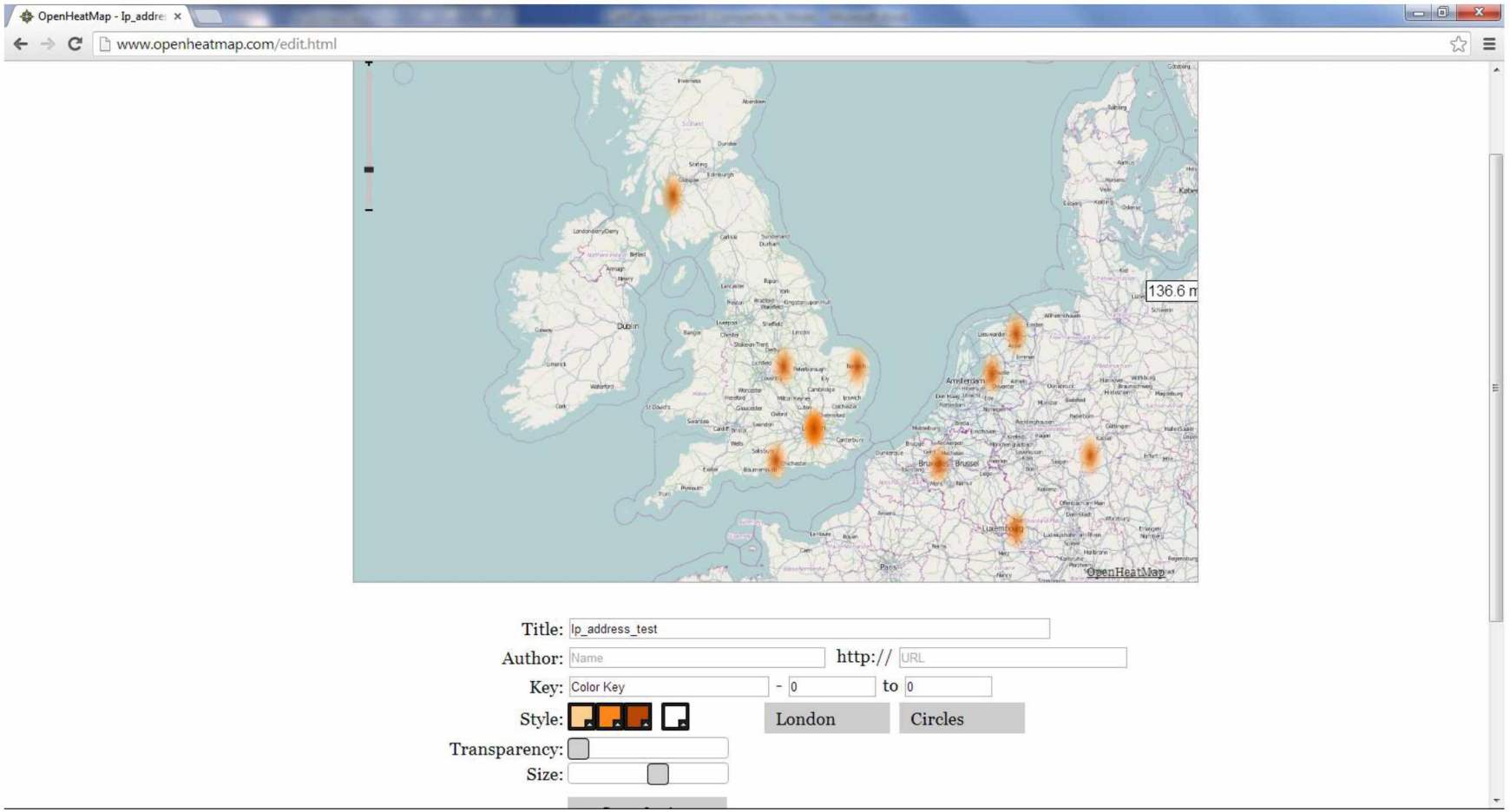
I changed the display theme to Manchester with White Borders. We can see that Canada and Sweden have some of the highest rankings while South Africa trails far behind. I'm not sure what statistics they used to determine one country was worse for women than another.



Finally I started back at the main site and accessed the sample file entitled IP Addresses.



There are only 85 data points in this file but OpenHeatMap can use IP addresses as location points. I changed the theme to London and zoomed in on the UK.



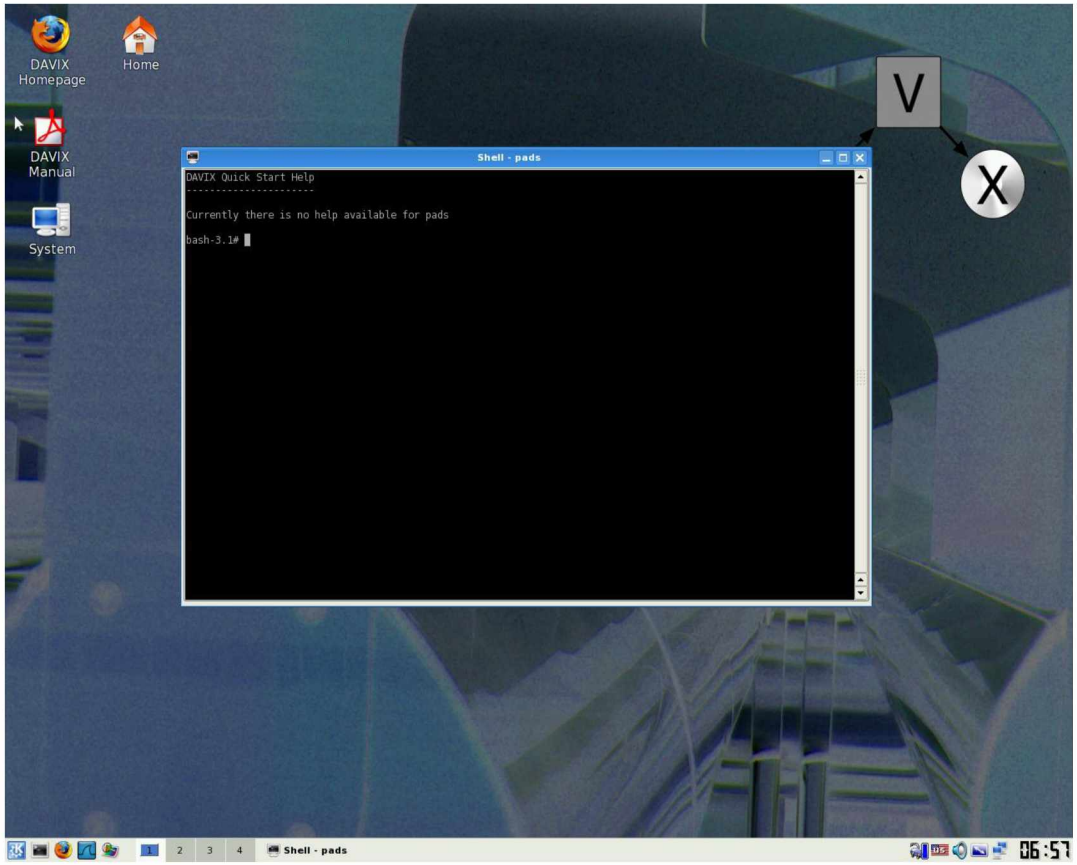
According to the documentation, OpenHeatMap understands a wide range of location information and can extrapolate that to display on a map. The list includes addresses, states, provinces, countries, IP addresses, counties, latitude and longitude, cities, and more. This makes it a very versatile and approachable means of viewing a heat map for a set of data. The inability to customize the view is probably its weakest point, along with it becoming much slower (and lags) in terms of zooming in and out of the map if a larger dataset is used.

Appendix E - Tool Example – PADS, Afterglow and GraphViz

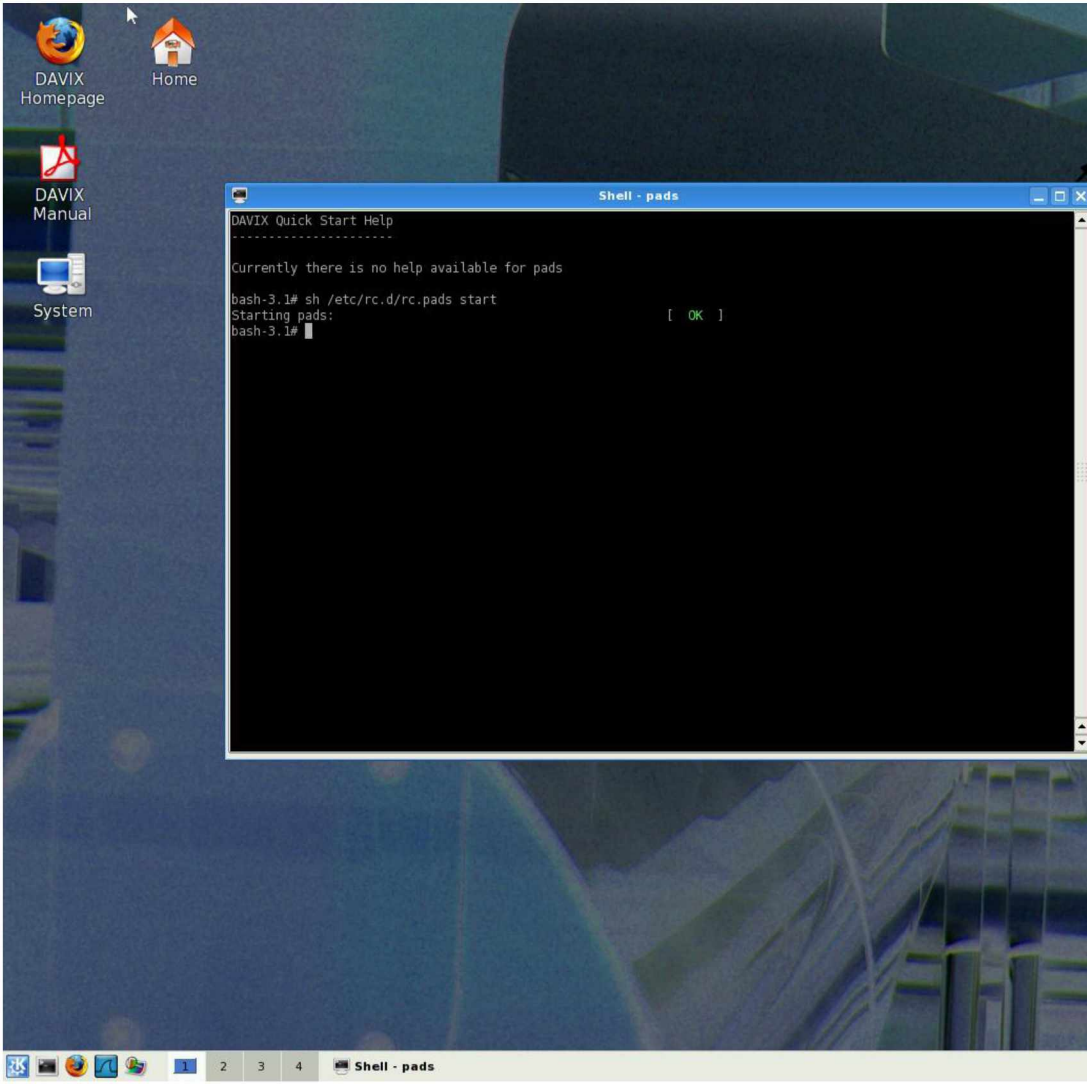
I evaluated PADS (Passive Asset Detection System) which passively tracks network traffic without sending packets; this information can then be converted using Afterglow to a format read by GraphViz. GraphViz can then generate a linked graph from the data.

This sort of network monitoring is valuable for identifying an insider threat. Insider threats are malicious acts of sabotage, information dispersal, and theft that can occur by individuals within the organization. By tracking all devices connected to the network the security analyst will be able to identify when rogue devices are attached which may result in data theft or other undesirable outcomes. (E.g., an individual bringing their personal laptop to their workplace with the intention of copying confidential files from network directories.)

I ran PADS from the DAVIX VM and opened it from the menu.

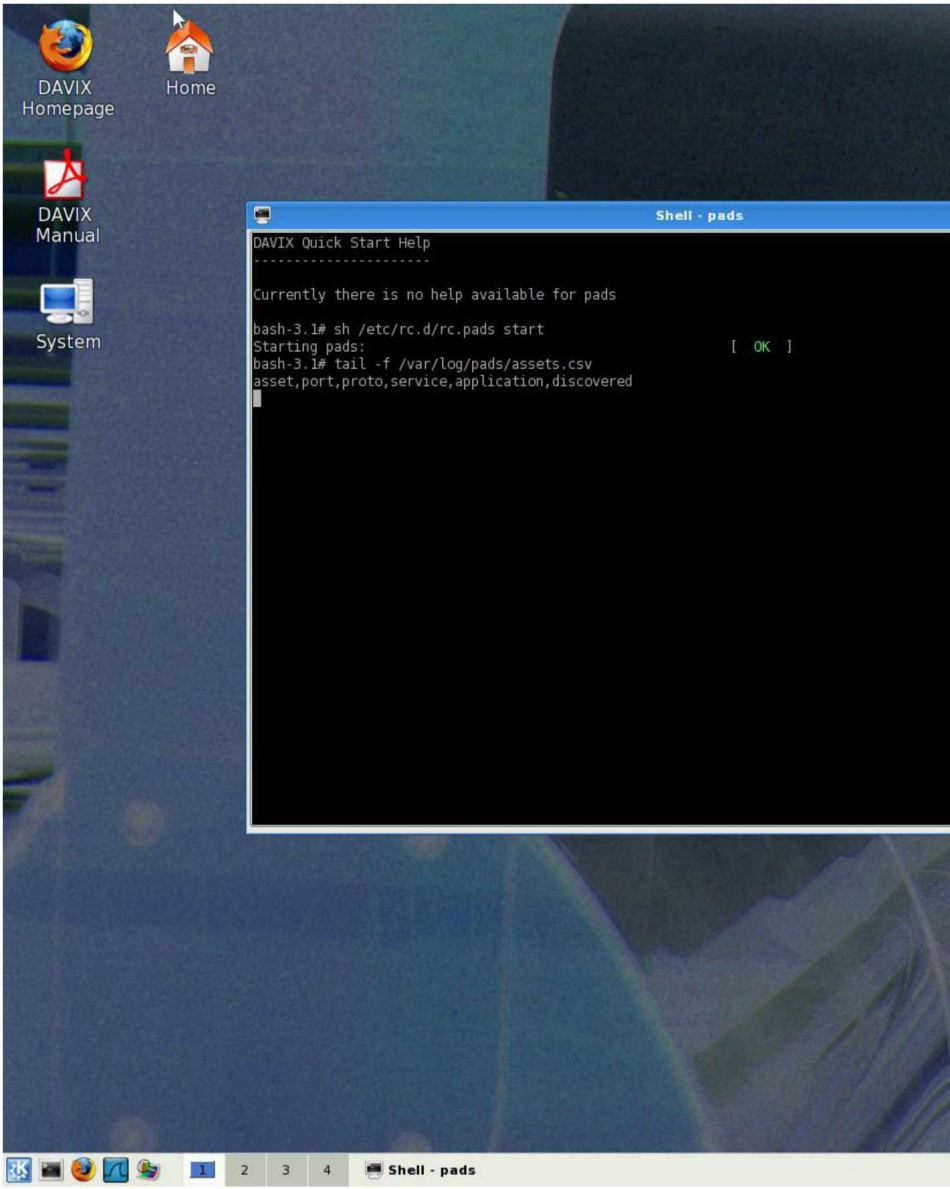


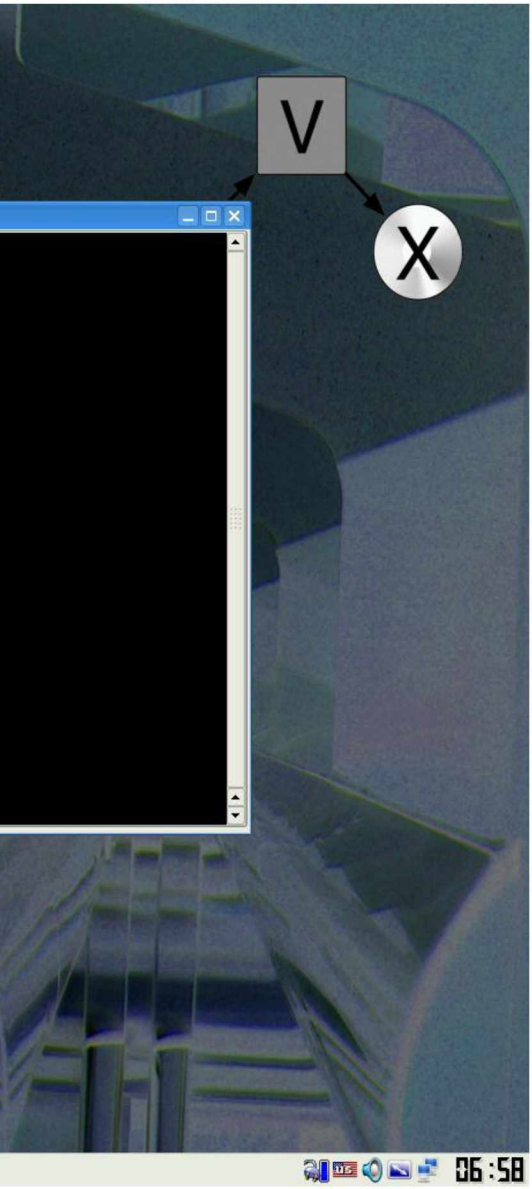
I then executed `sh /etc/rc.d/rc.pads start` to start the tool.



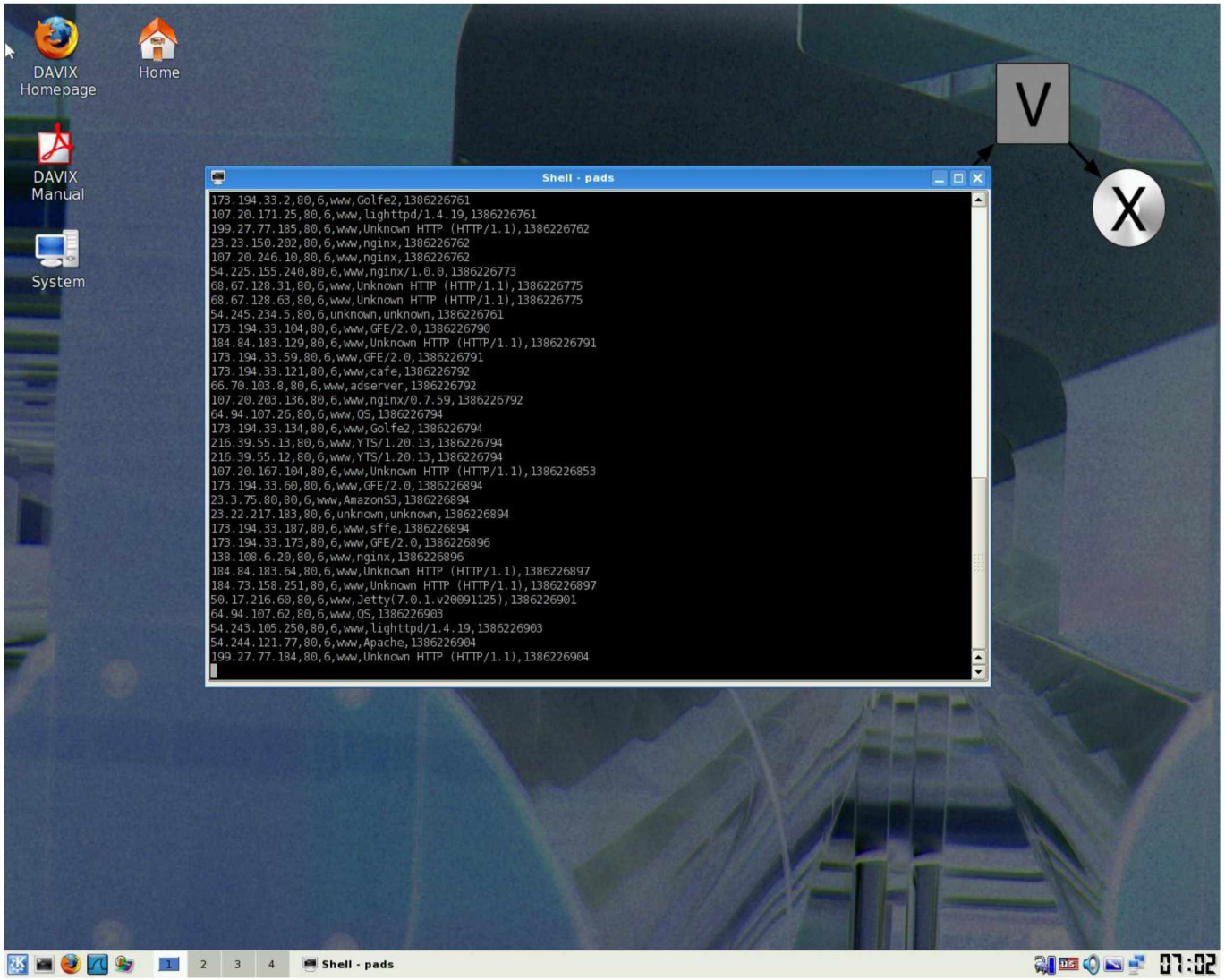

V
X
06:58

I set the log to append to the tail of the assets.csv file.



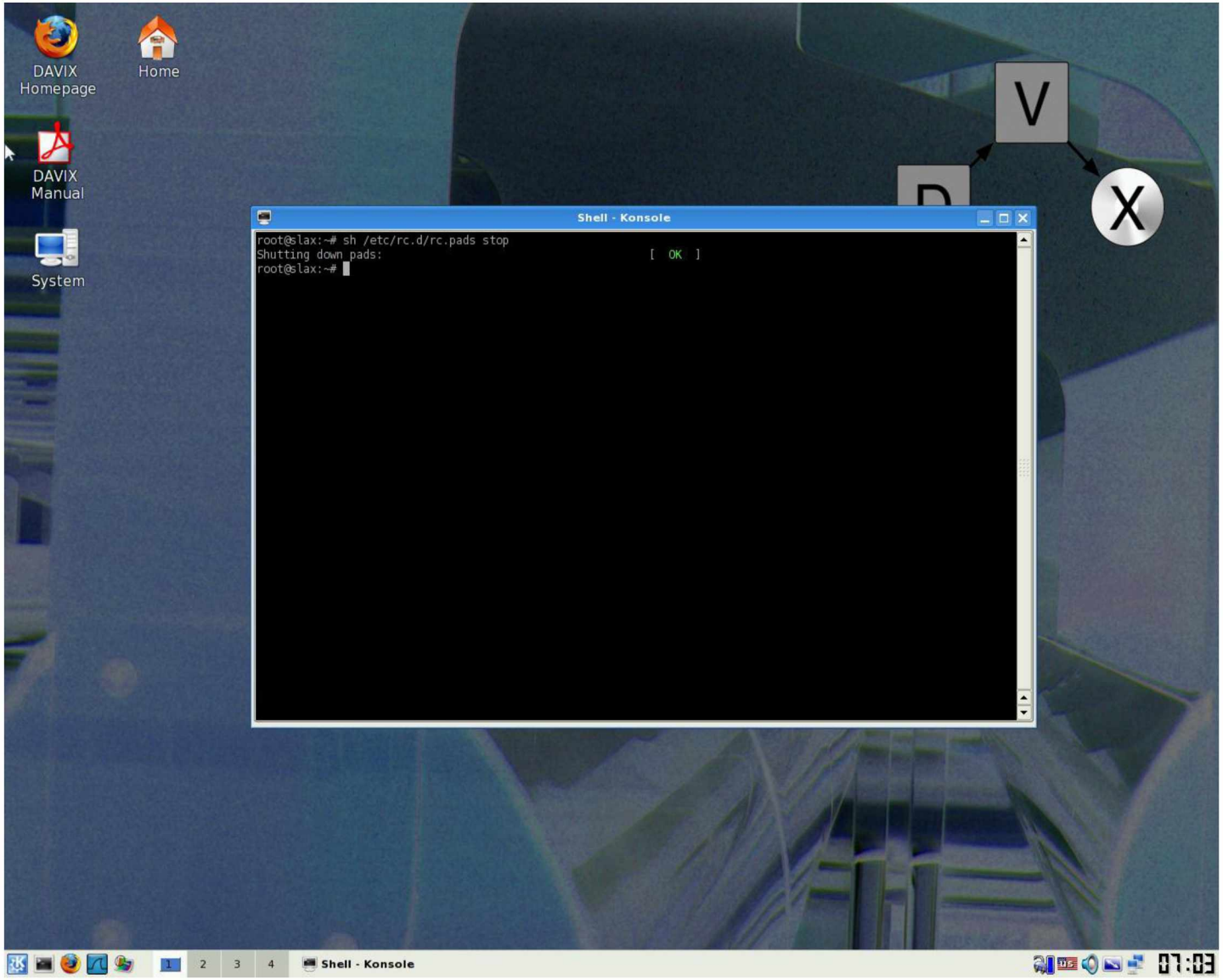

V
X
06:58

I then opened Firefox and began browsing the web. Data began appearing in the console.

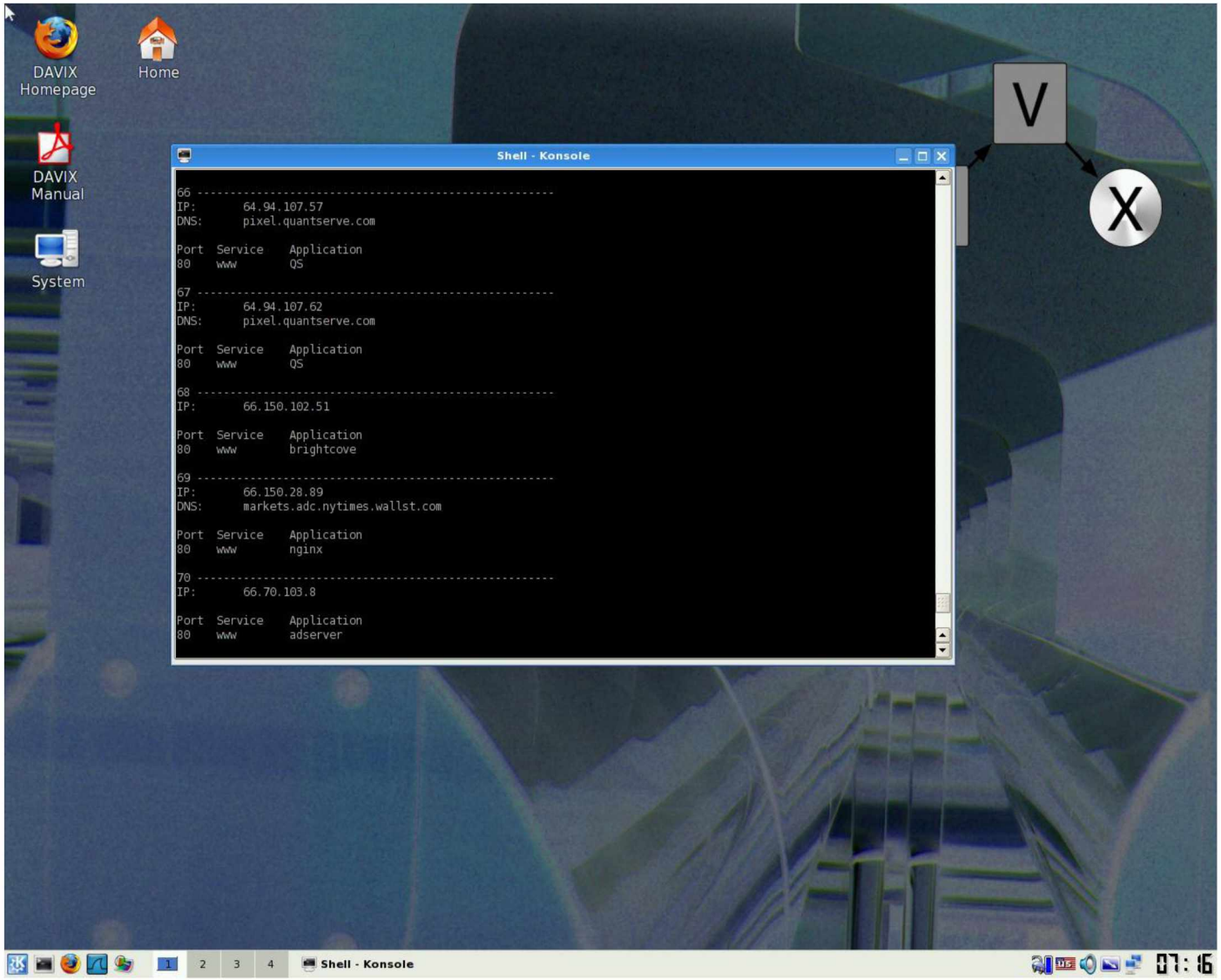


120

After I had browsed enough I shut down pads by executing `sh /etc/rc.d/rc.pads stop`



I took a look at the pads-report which presented a cleaner view of the data.



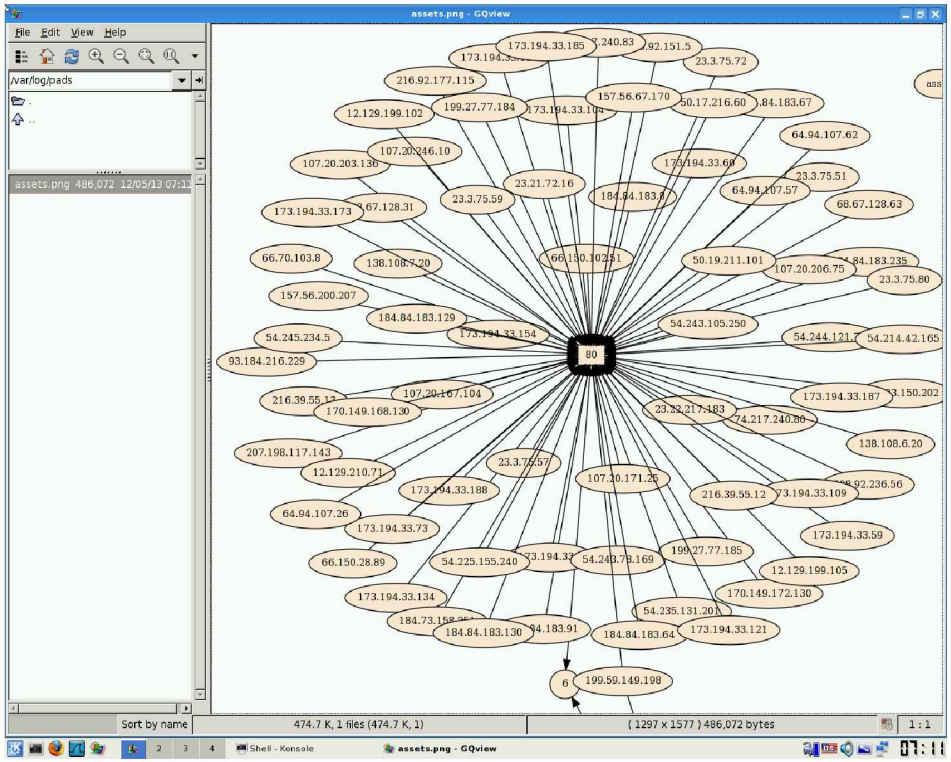
Then I used Afterglow to convert the csv file to a GraphViz dot file by executing the following command:

```
cat assets.csv | afterglow.pl > assets.dot
```

Then I rendered the assets.dot into a gif file as follows:

```
neato -Tpng -o assets.png assets.dot
```

Then I opened the gif file with the command `gqview`. I was presented with a linked graph displaying all device IPs that had been logged in the assets file.



This can theoretically be used to provide a quick visual view of any new devices that connect to the network, although the process is a little convoluted. My demonstration of PADS and the graphical representation of devices is a modest attempt at insider threat detection and only one of many techniques a security specialist may use.

In Applied Security Visualization a great deal of thought is put into the idea of assessing users and their precursors, which are activities that typically fall outside normal job requirements and can be actively scored in severity. For example, turning off one's AV software may be a precursor with a high risk score whereas browsing Craigslist Jobs

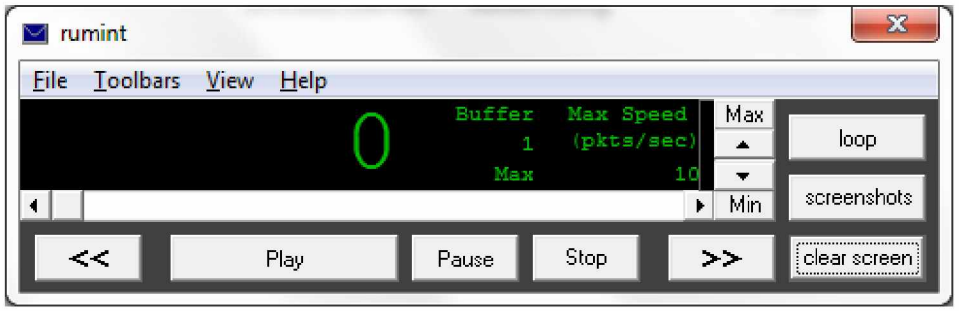
may be a lower risk score. An even higher risk would be a former employee attempting to access the network. Once precursors have been identified they can be used to more effectively catalog the network traffic within an organization and visualization tools can be used to help identify individuals who may pose an insider threat.

Establishing organizational-level network monitoring is often done with a Security Information and Event Management (SIEM) solution which permits detailed monitoring of network traffic and the establishment of alerts and other features to prevent not only insider threats but other security threats. Unfortunately most are full-scale solutions available for purchase and could not be tested for academic purposes.

Appendix F - Tool Example – rumint

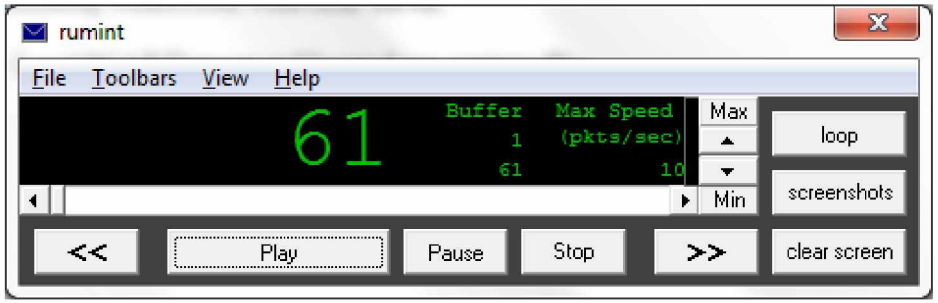
rumint v2.14 allows visualization of network captures. It is available at http://www.rumint.org/

I ran rumint and went to the main screen.



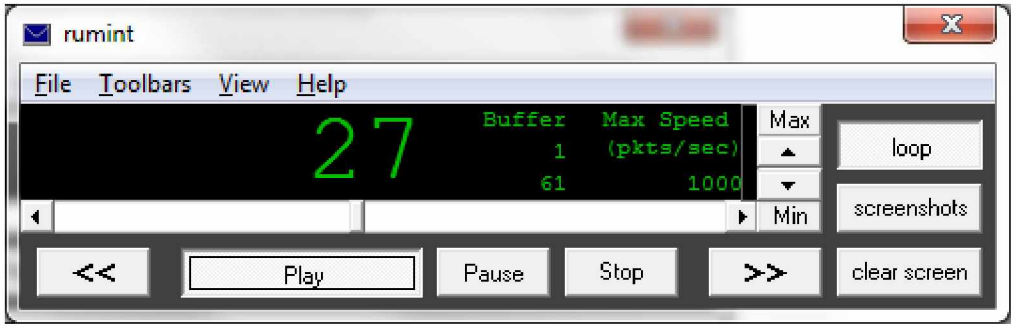
I decided to use the Wireshark Sample Capture list located at http://wiki.wireshark.org/SampleCaptures and selected the same file I had used when evaluating Wireshark. This was unistim_phone_startup which is described as "Shows a phone booting up, requesting ip address and establishing connection with cs2k server."

I went to File -> Load PCAP Dataset and located my capture file.

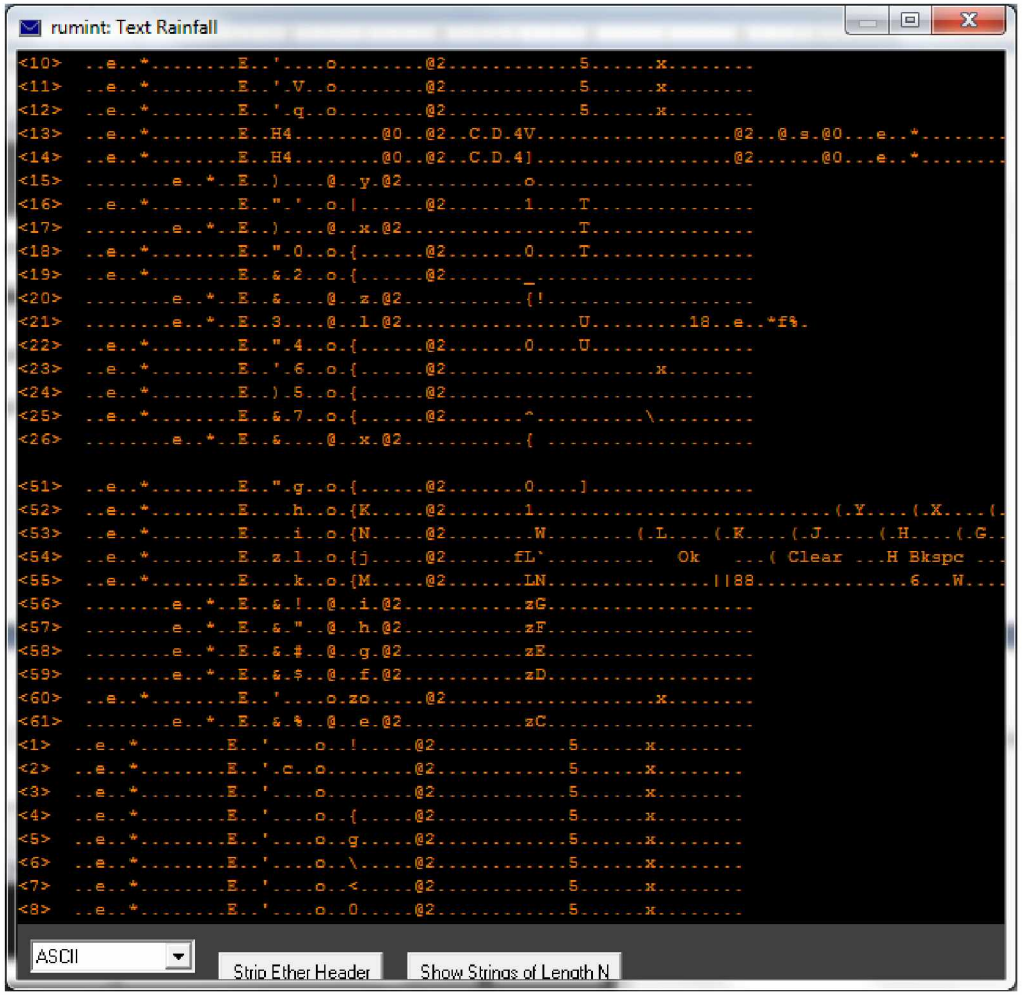


This provides me with the same number of packets as viewed via Wireshark.

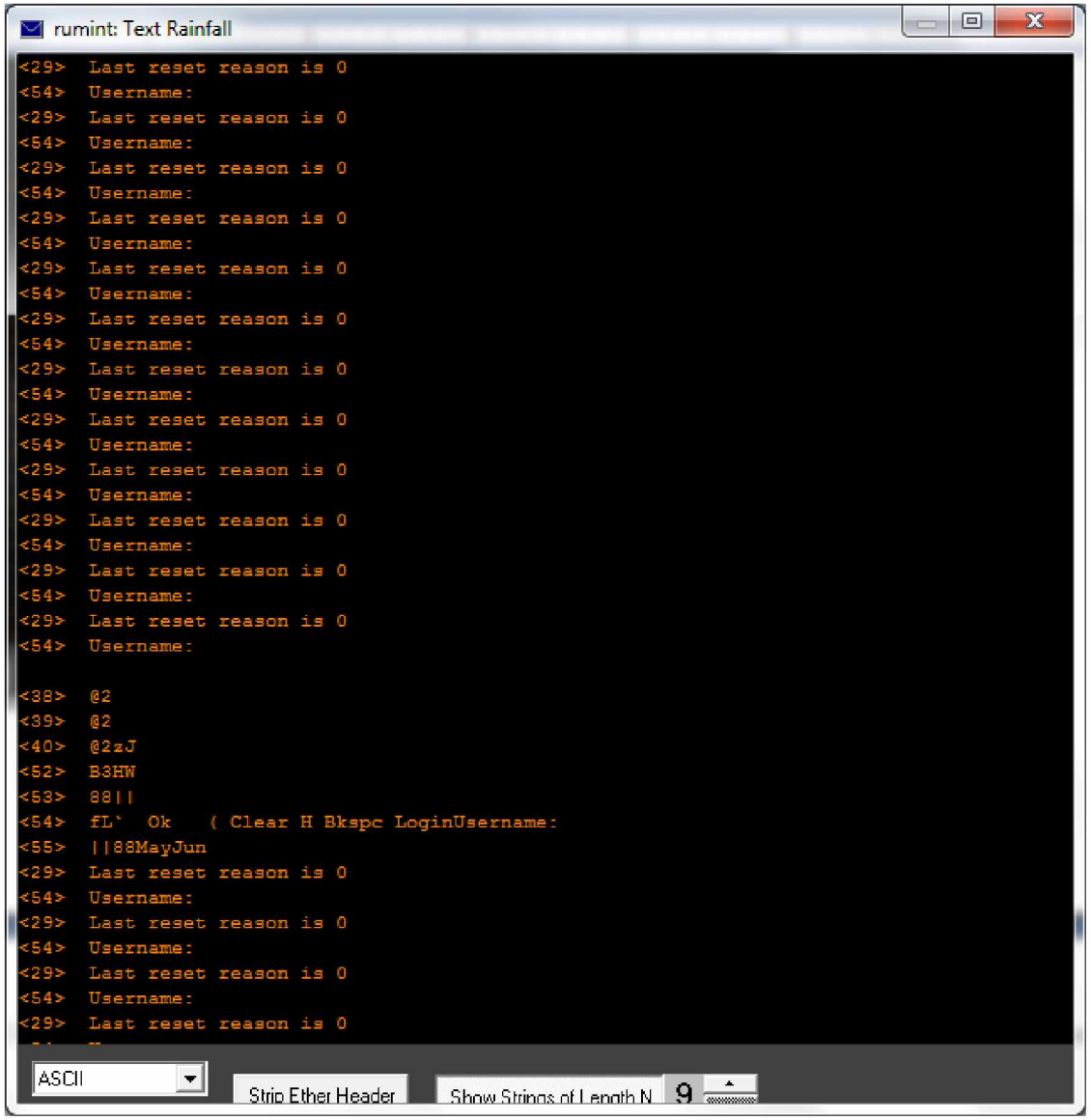
I selected the Loop button and then hit Play; the capture began running.



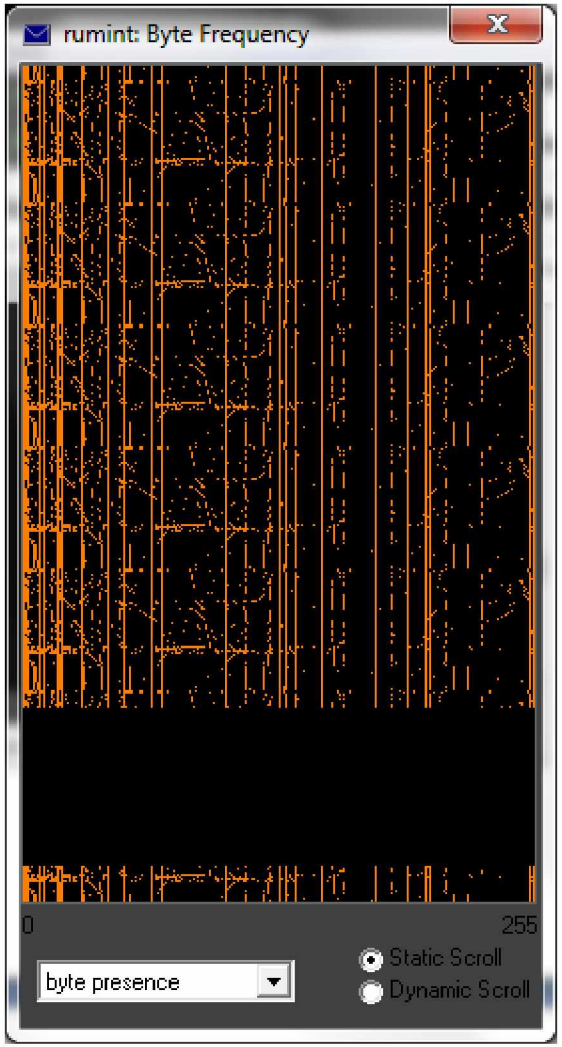
I went to View -> Text Rainfall and entered the Matrix.



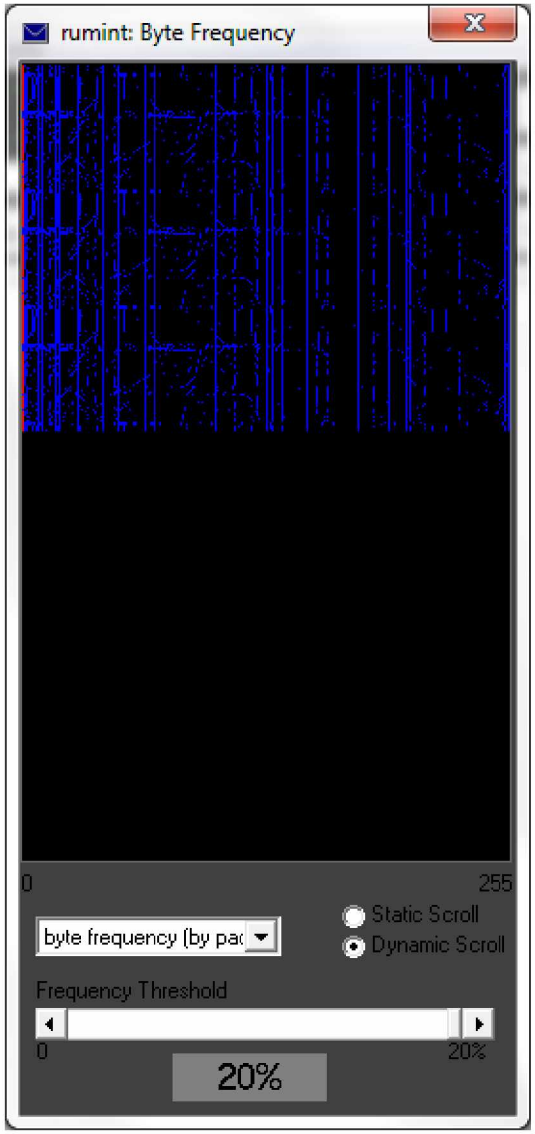
I switched the Show Strings of Length N to 9. It began showing text descriptions.



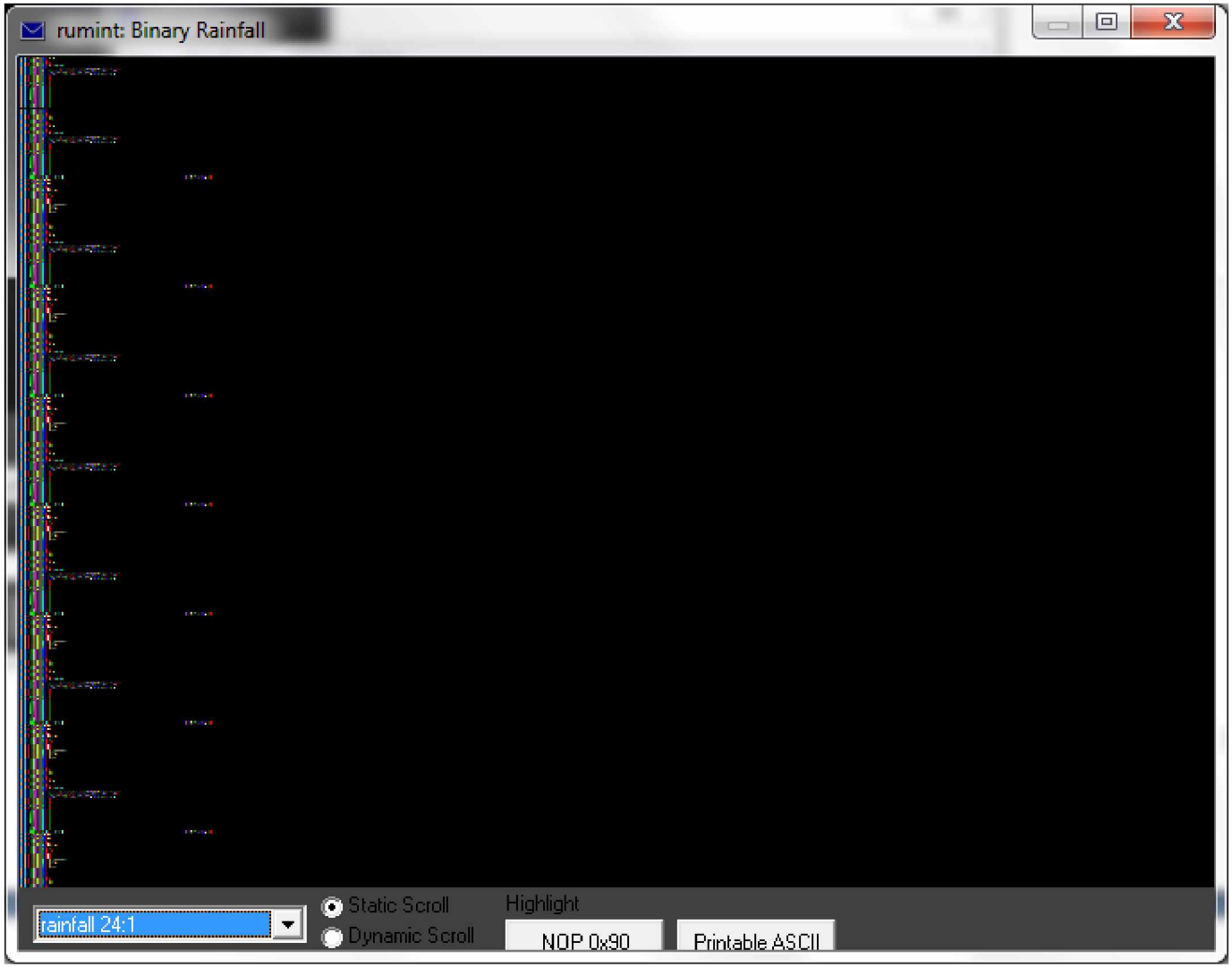
I closed the window. I went to View -> Byte Frequency and was shown this dynamic visualization.



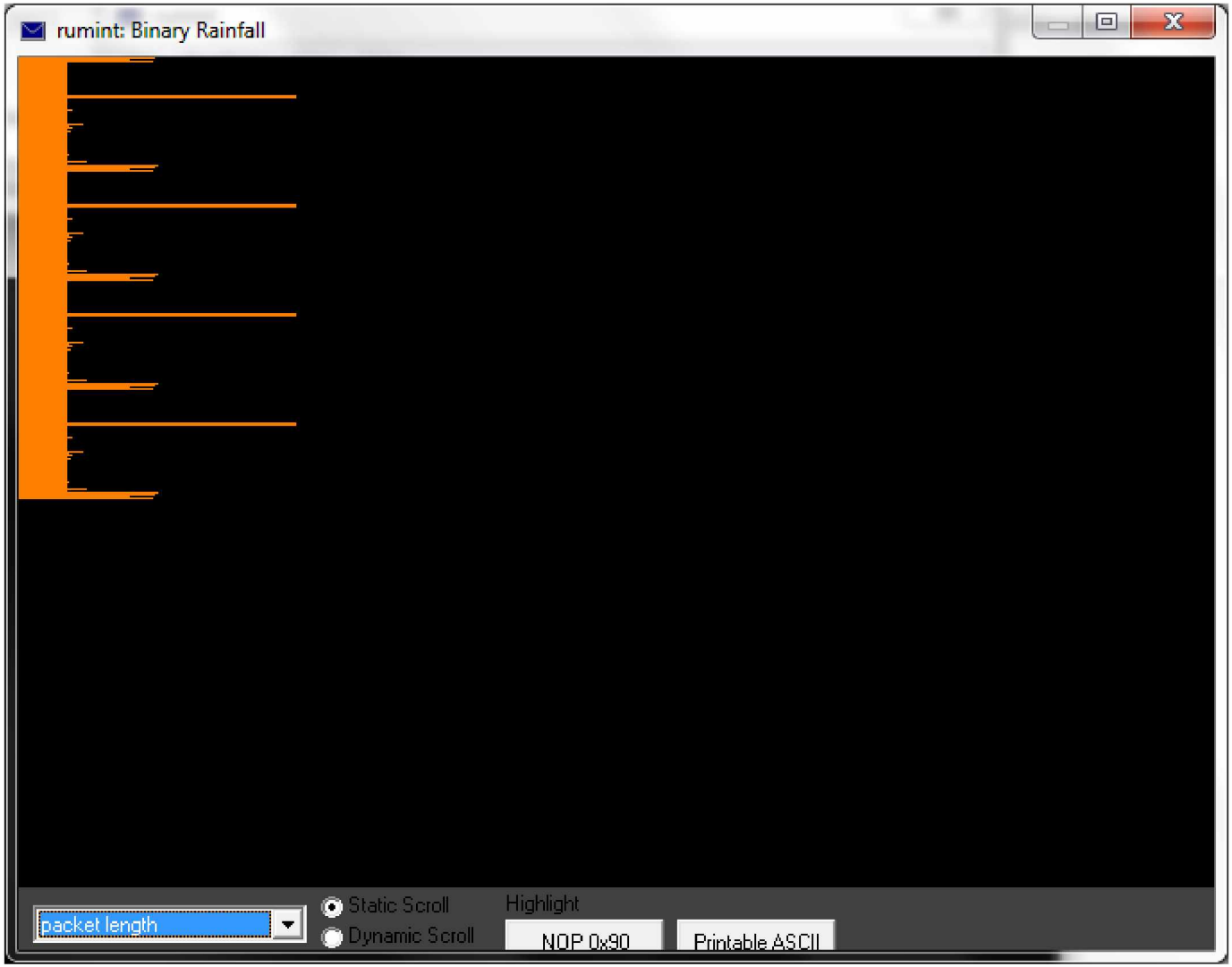
I switched to byte frequency by packet, Dynamic Scroll, and set the Frequency Threshold to 20%.



I closed the window and went to View -> Binary Rainfall. I switched it to “rainfall 24:1”.

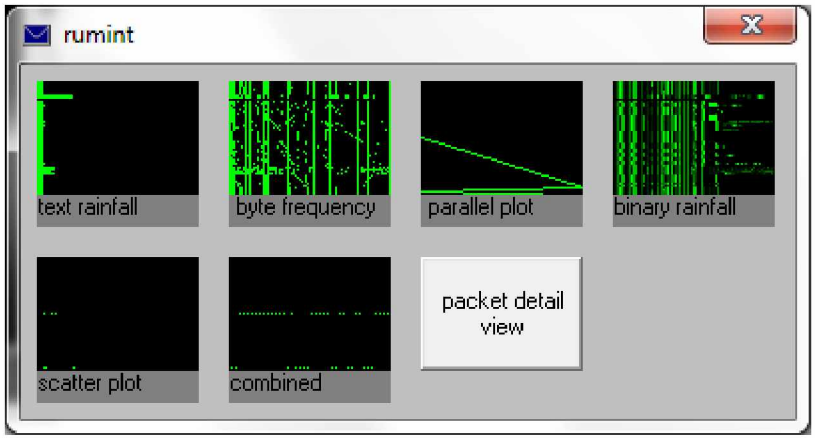


Then I switched it to packet length.



I closed the window and went to View -> Detail.

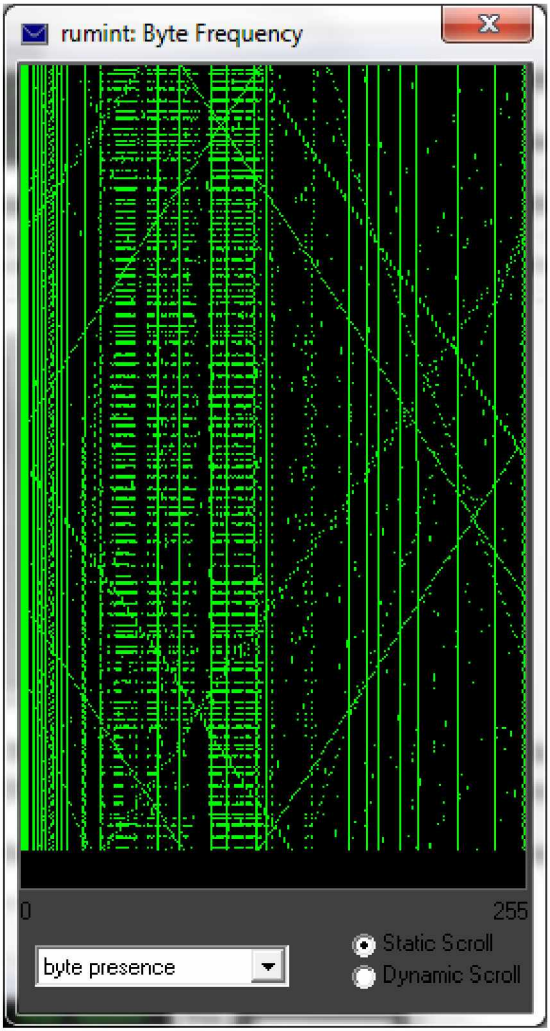
I closed the window and went to View -> Thumbnails. This summarized all of the available graphs and they were all updating dynamically. Clicking on one of the thumbnails would open the graph itself.



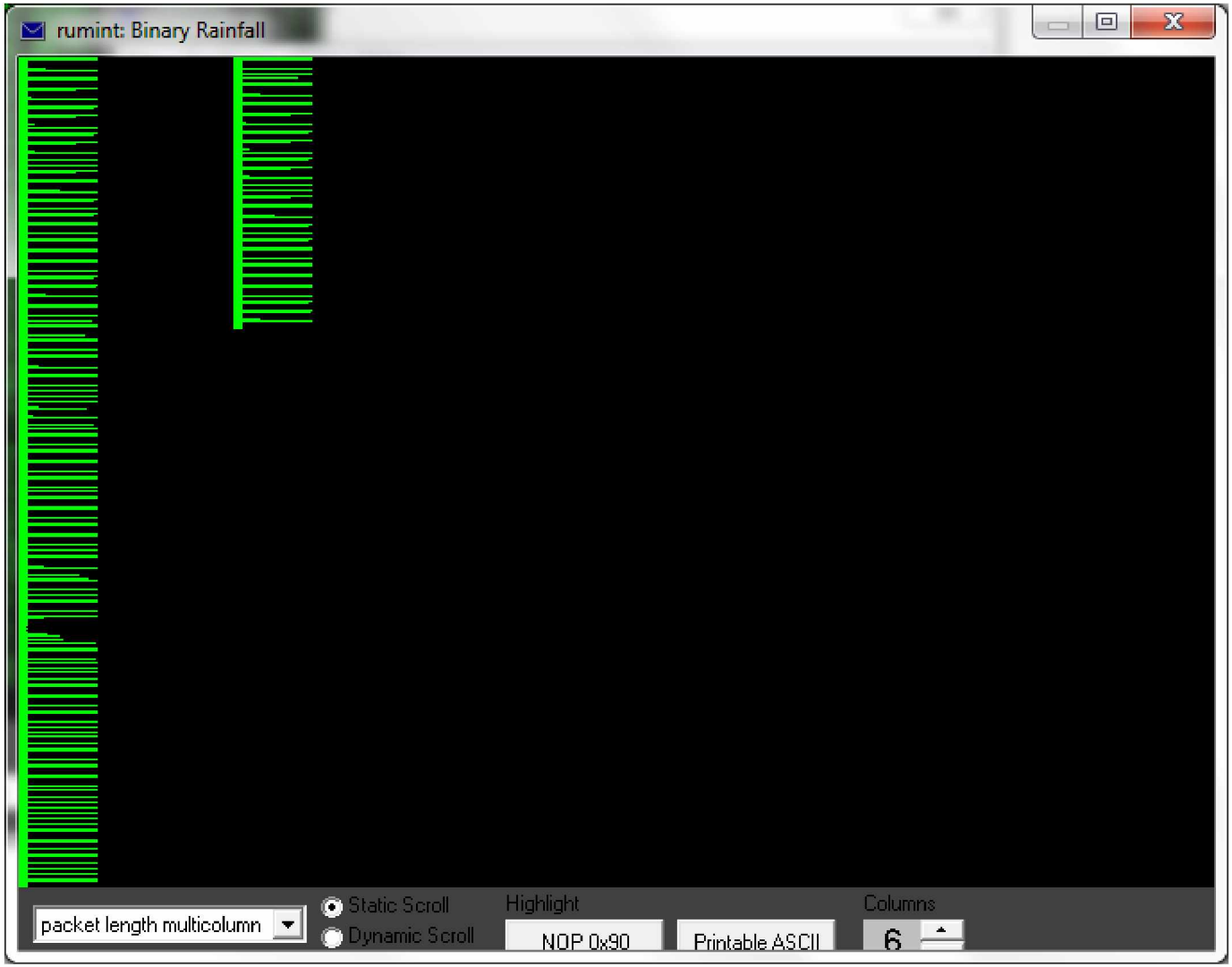
I decided to download and test another capture. This was tcp-ecn-sample described as "A sample TCP/HTTP of a file transfer using ECN (Explicit Congestion Notification) feature per RFC3168. Frame 48 experienced Congestion Encountered."

I went to File -> Load PCAP Dataset and opened the tcp-ecn-sample. This had significantly more packets.

I hit Play and went to View -> Byte Frequency.



I closed the window and went to View -> Binary Rainfall. I had Packet Length Multicolumn selected.

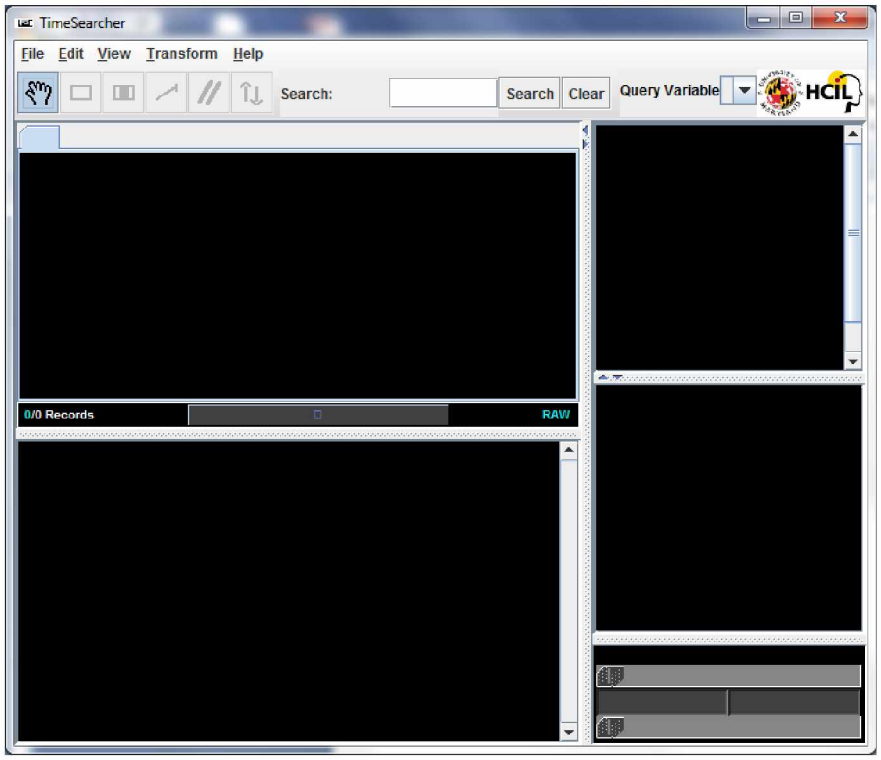


This is an interesting tool for replaying network captures and visualizing the associated data in a variety of ways.

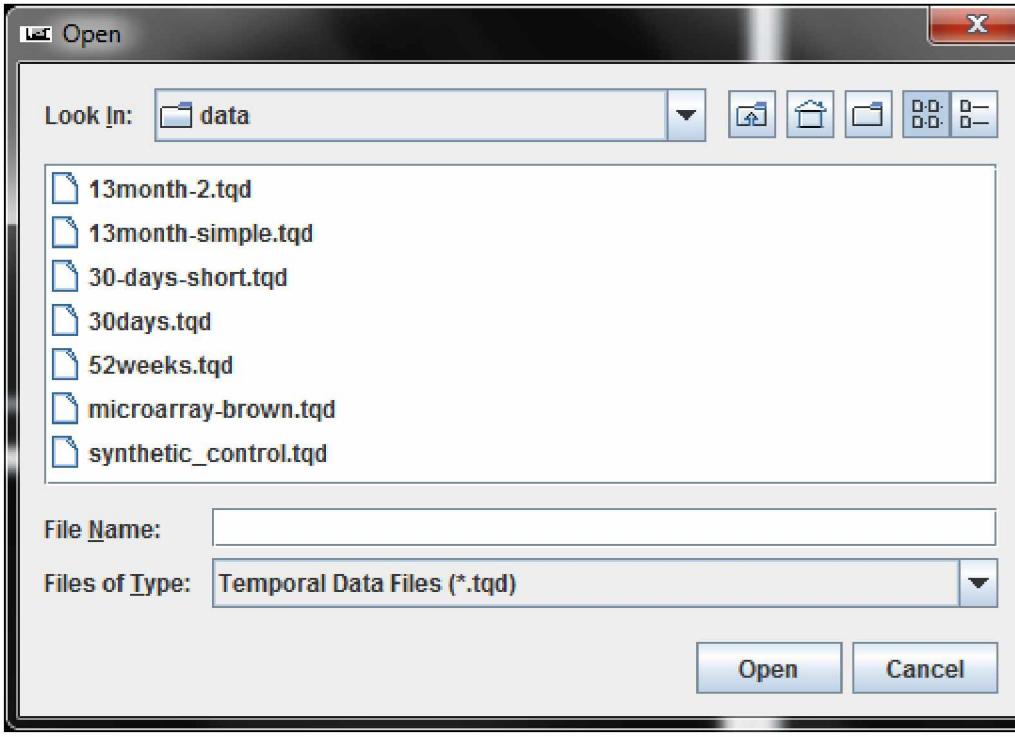
Appendix G - Tool Example – Timesearcher 1

Timesearcher 1 allows analysis of time series data. It is available at http://www.cs.umd.edu/hcil/timesearcher/

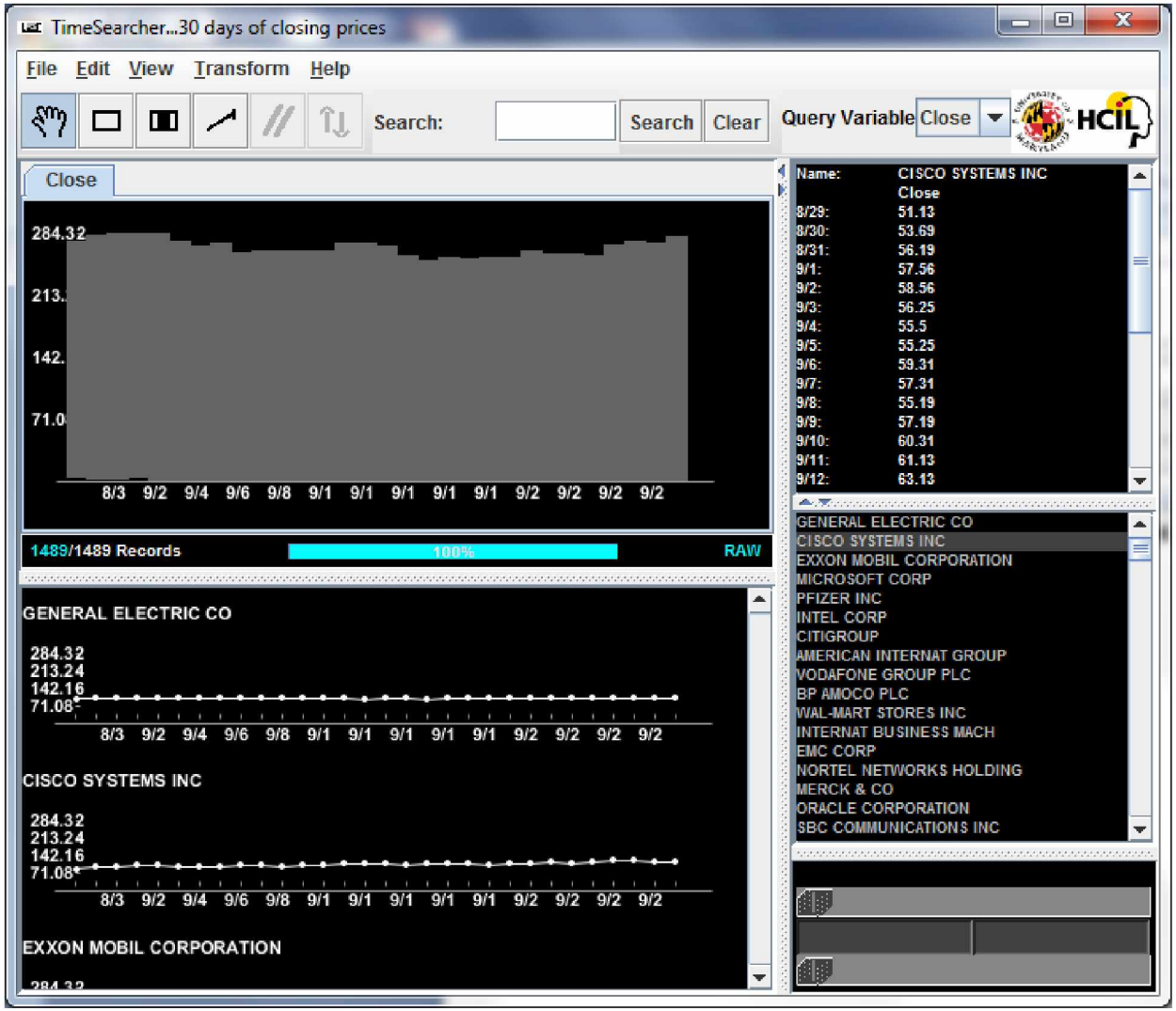
Timesearcher 2 is also available but when I attempted to test it the application kept freezing when I tried to open a data file, therefore I will be reviewing Timesearcher 1.3.7. I ran the application via the ts batch file.



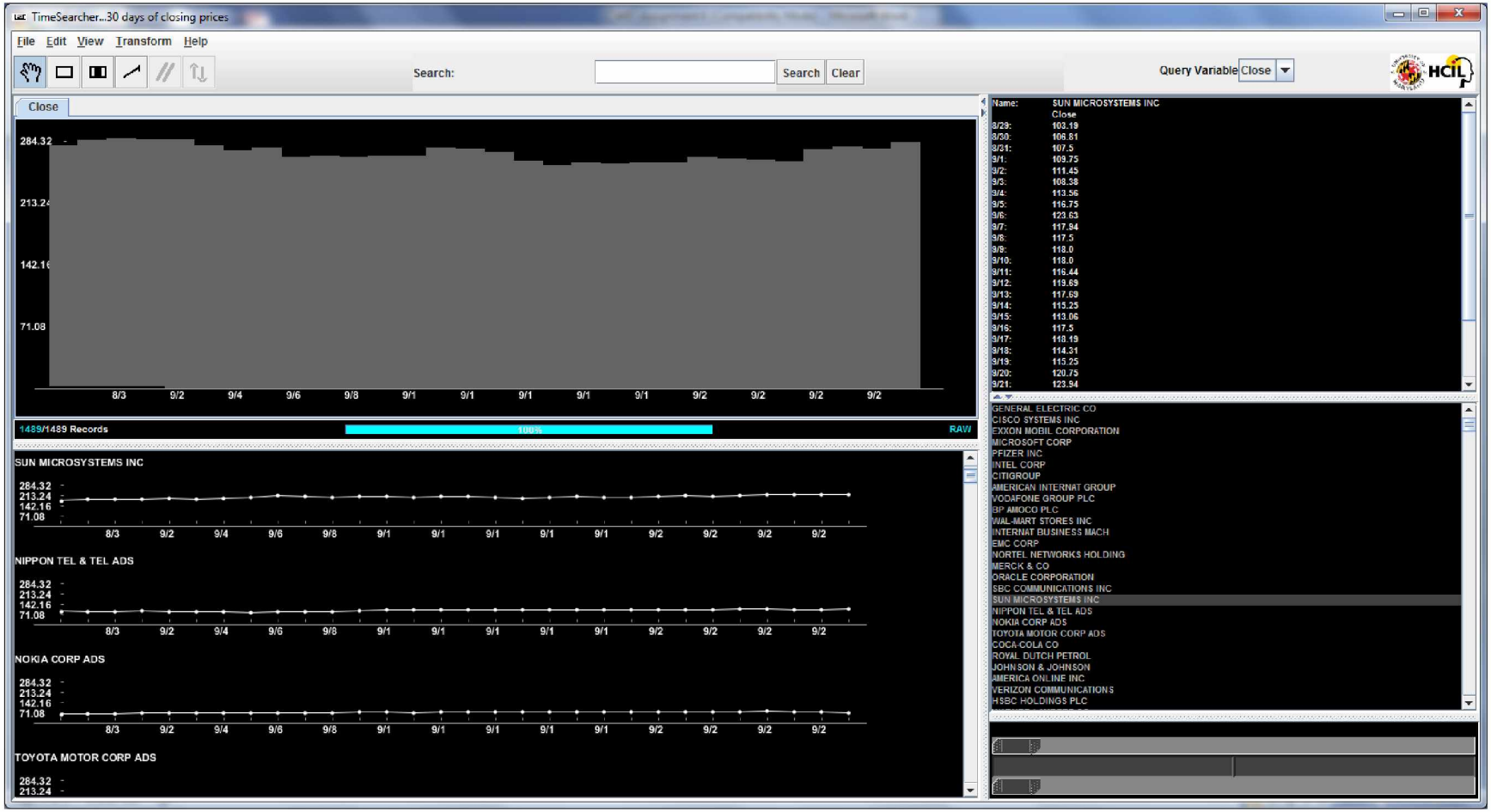
I then went to File -> Open Data File …



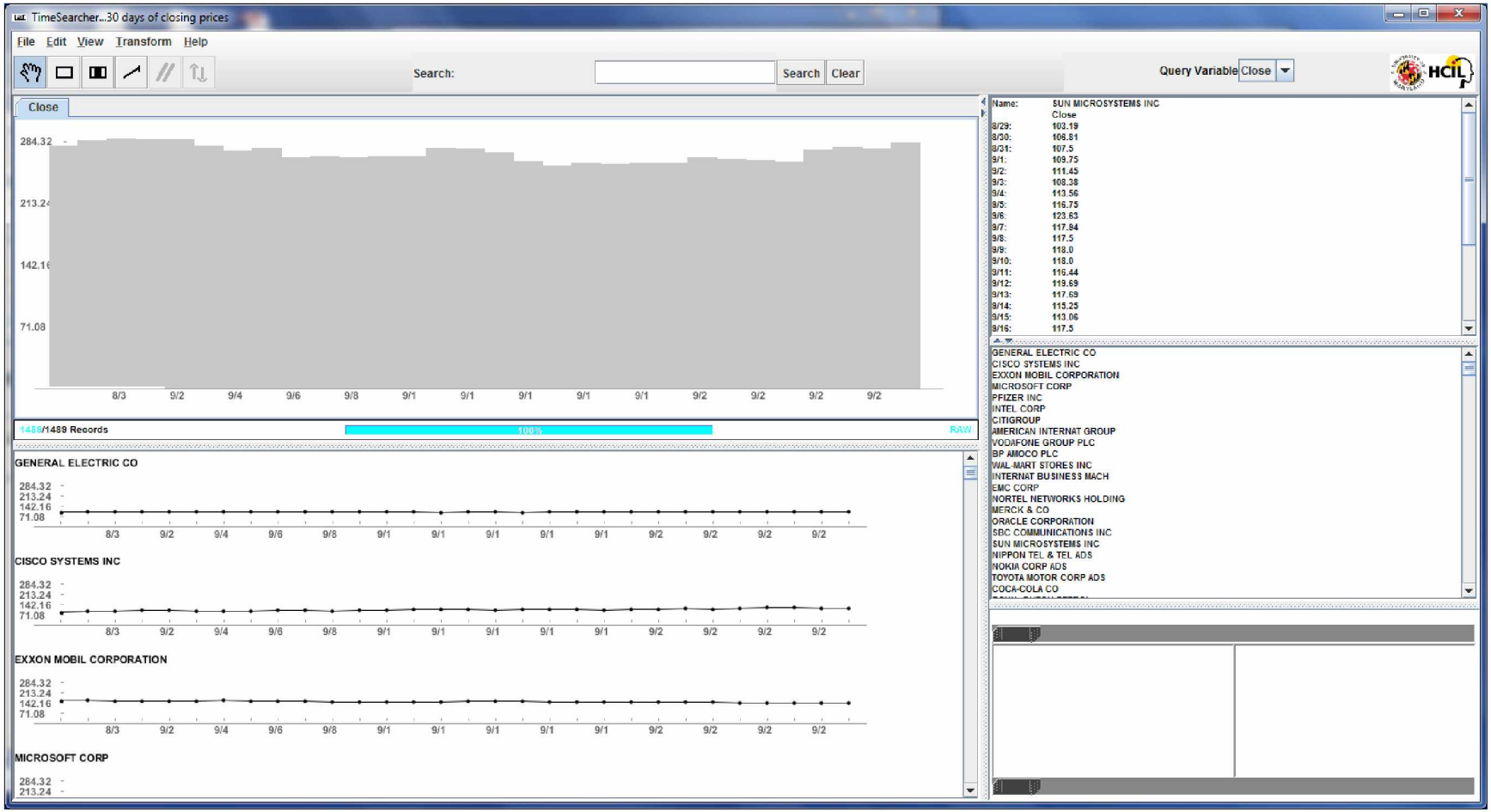
I selected 30days.tqd and clicked Open. This file shows 30 days of closing prices for varying stocks.



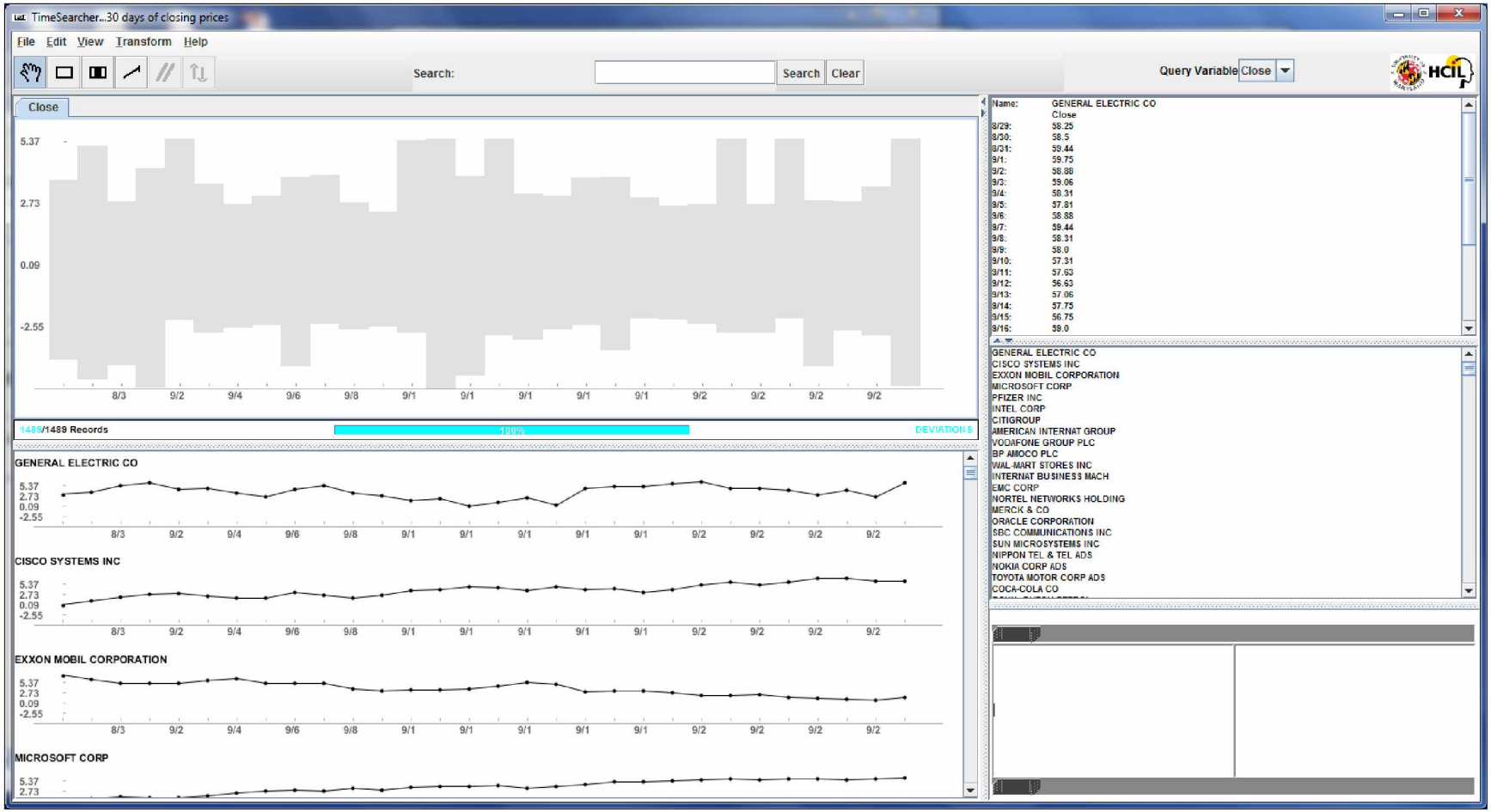
The graph in the upper left displays the summary information but by clicking on the specific data points in the list on the lower right I was able to jump to graphs for each stock. For example, I selected Sun Microsystems. The upper right frame displays a list of the closing prices for Sun and the lower left frame has jumped to the Sun graph.



I went to View -> Invert Colors.

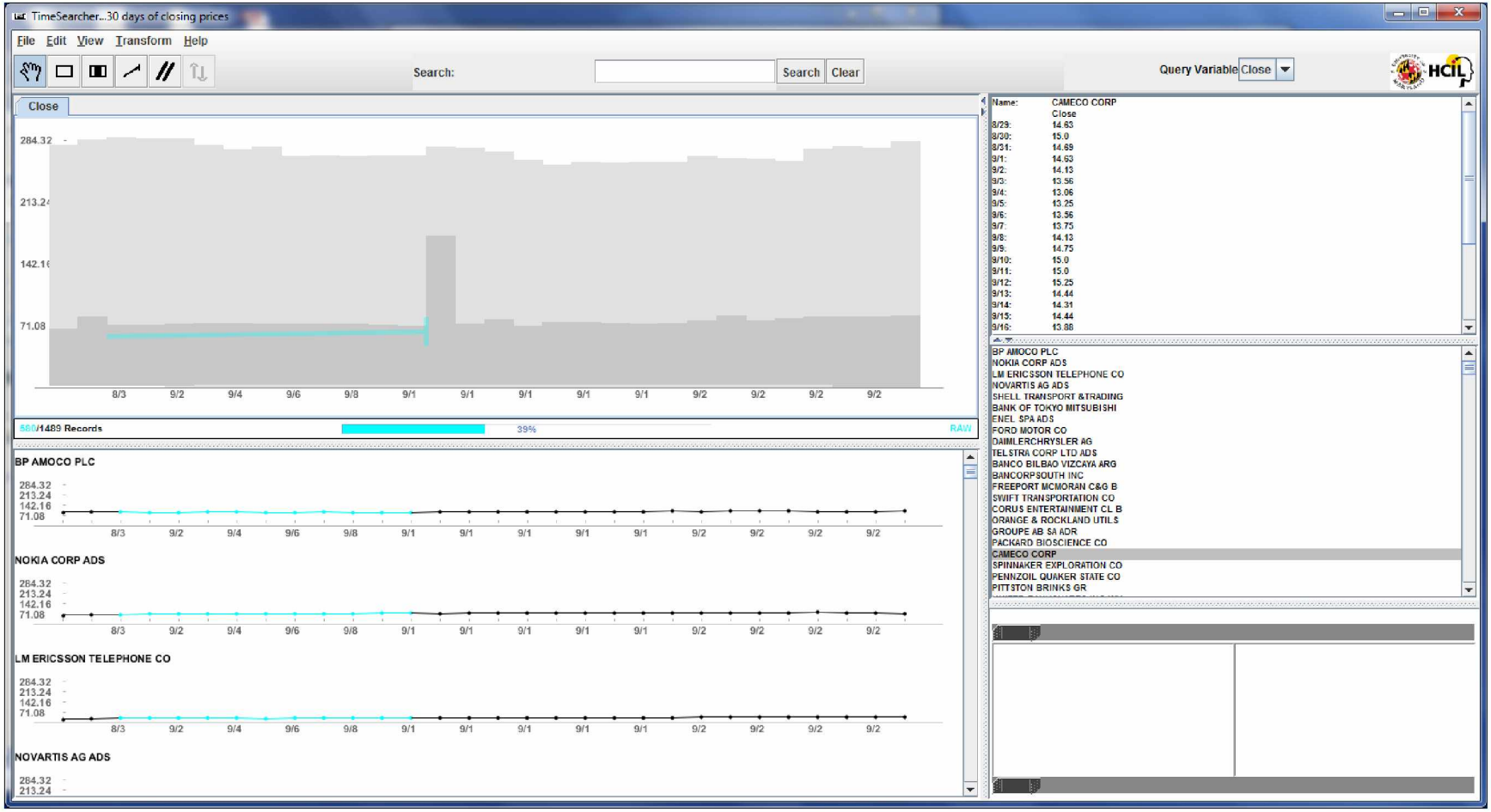


I then went to View -> Deviation Normalized.

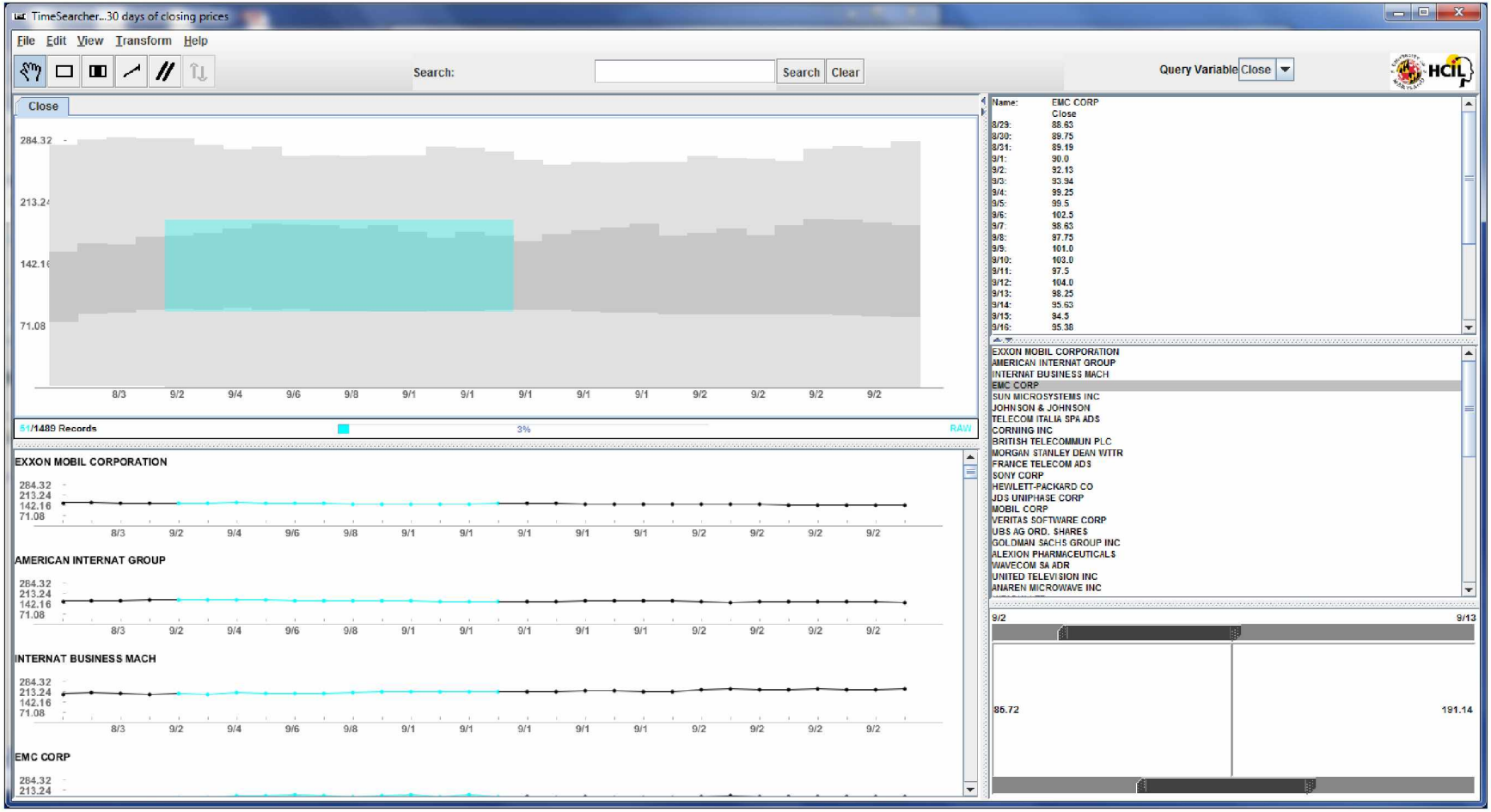


I switched it back to View -> Raw Data. Then I selected the "Draw Angular Query" icon and drew a line in the graph.

This limited the stocks to those with a closing price below my line within the selected date range.

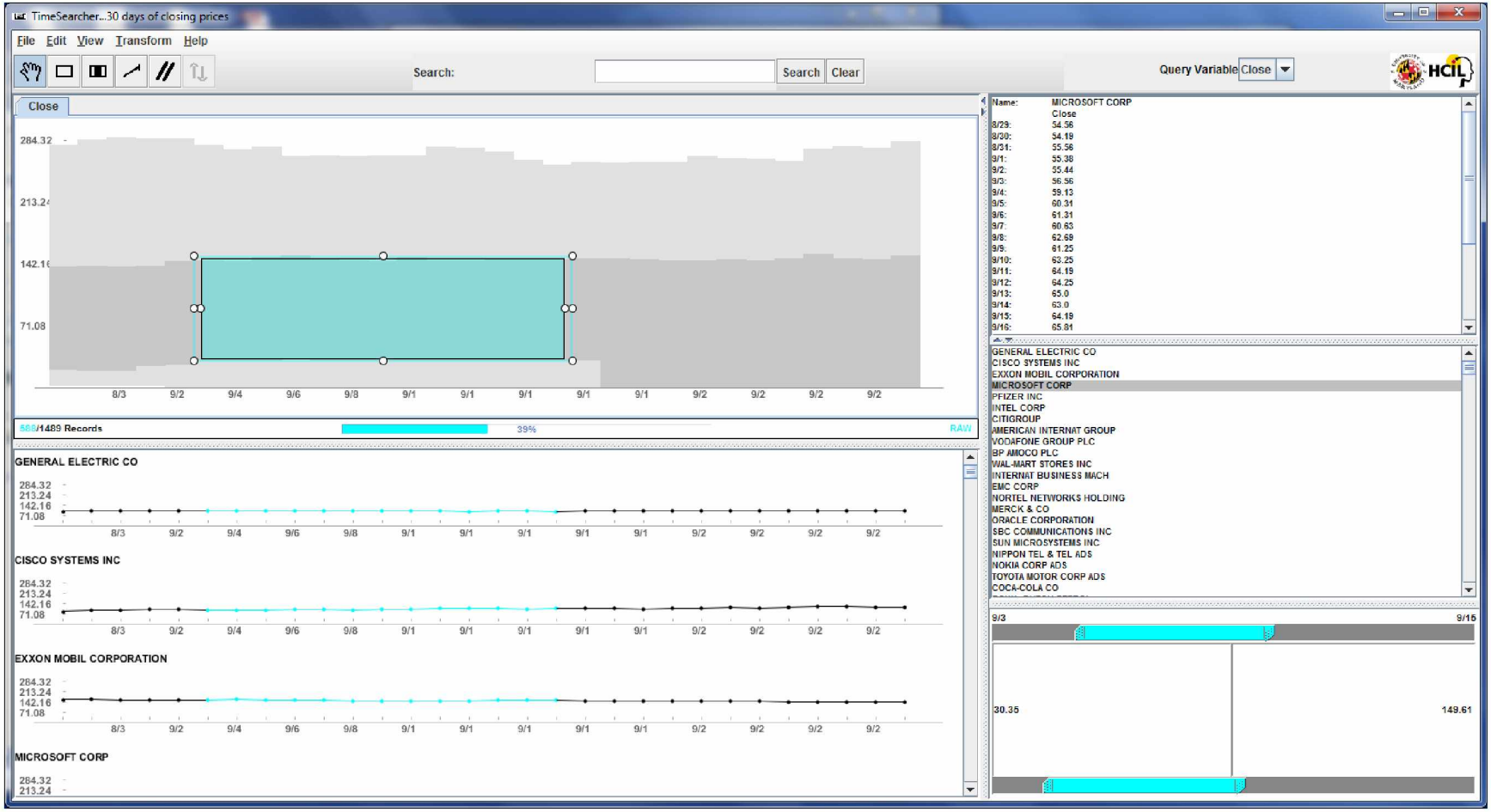


I went to Edit -> Clear Queries. Then I selected “Draw Timebox” icon and drew a box in the graph. This limited the stocks to those with values within the box during the specified time frame.

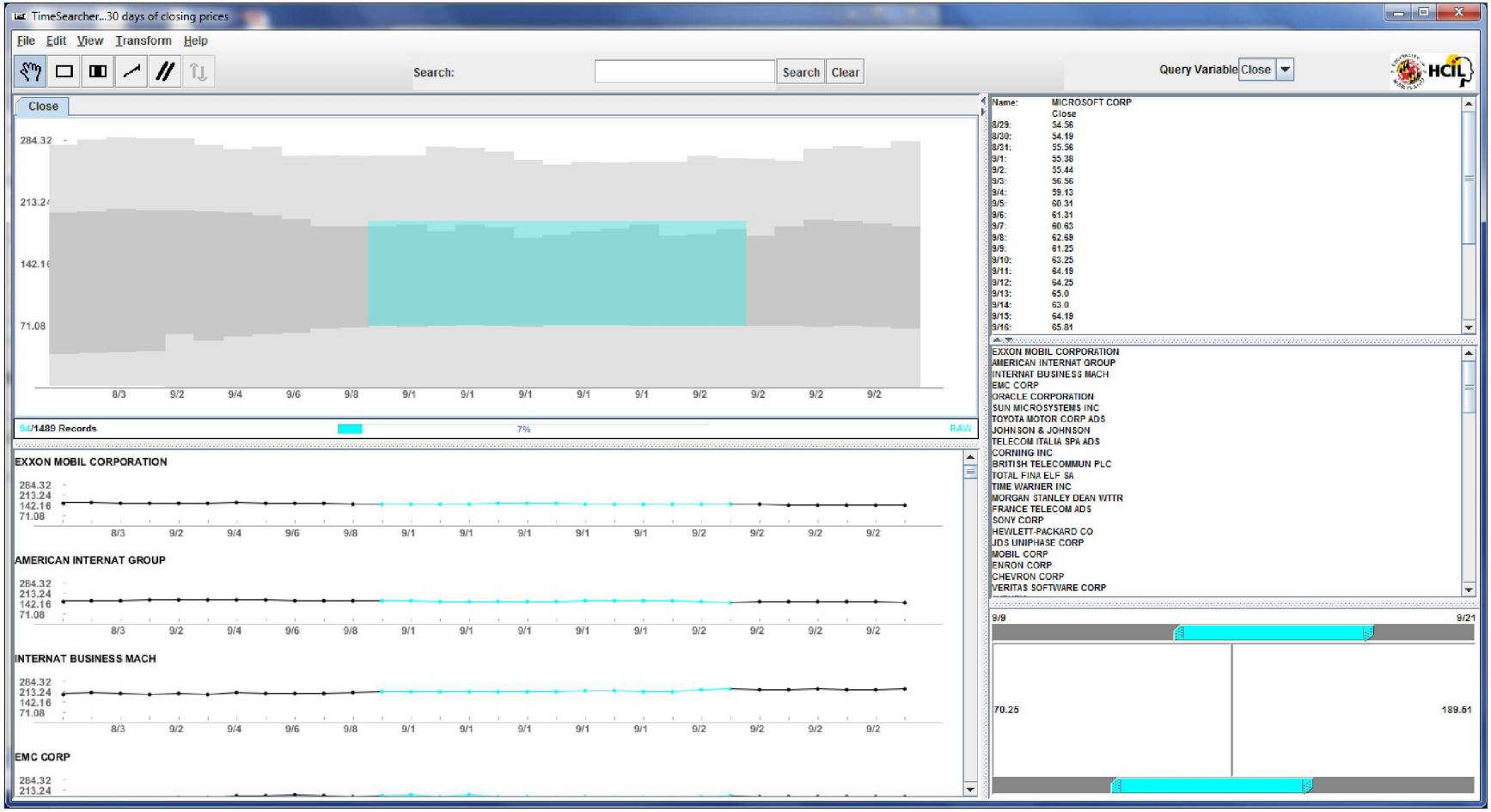


I went to Edit -> Clear Queries. Then I selected the "Draw Variable Time Timebox" icon and drew a box in the graph.

This limited the stocks to those with values within the box during the specified time frame.

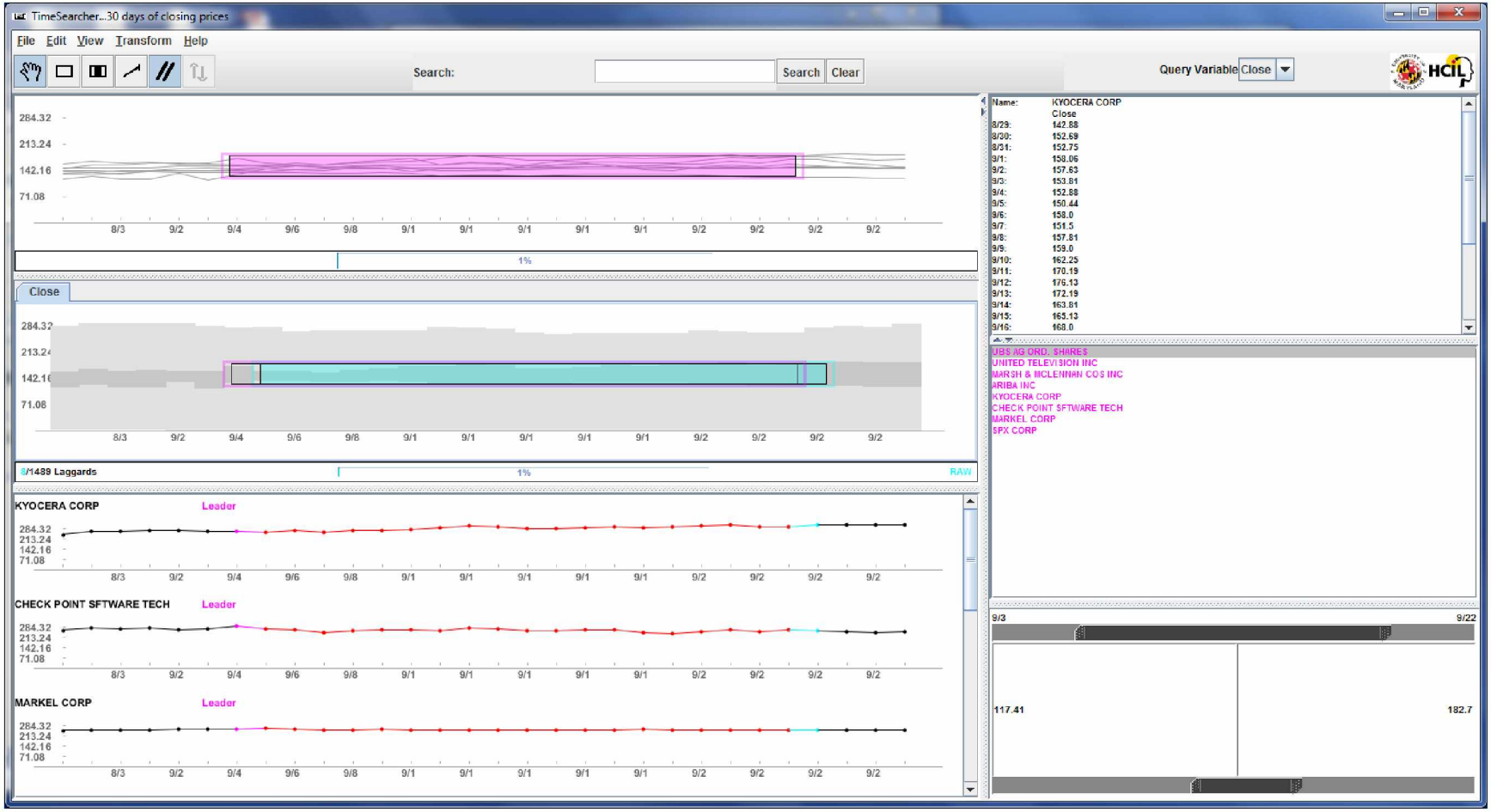


Since this is a variable time box I can then use the sliders in the lower right corner to move the time box. Note that the percentage of records bar is showing the number of data points that fall within the selected area of the graph.

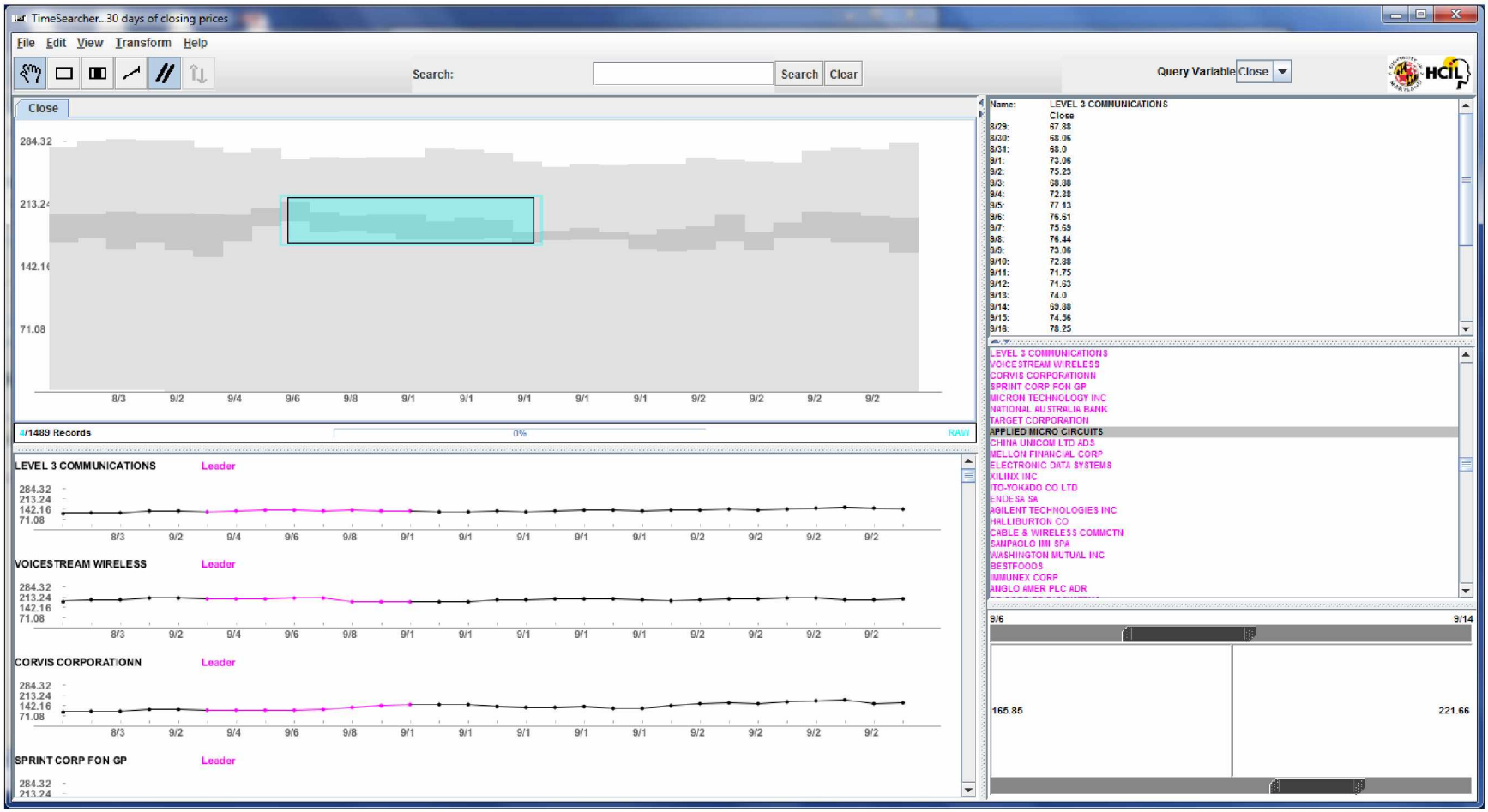


I went to Edit -> Clear Queries and then drew a new variable time box and selected the "Leaders and Laggards" icon.

This time box is only showing leaders. Note how the list of stocks to the right has changed appropriately.



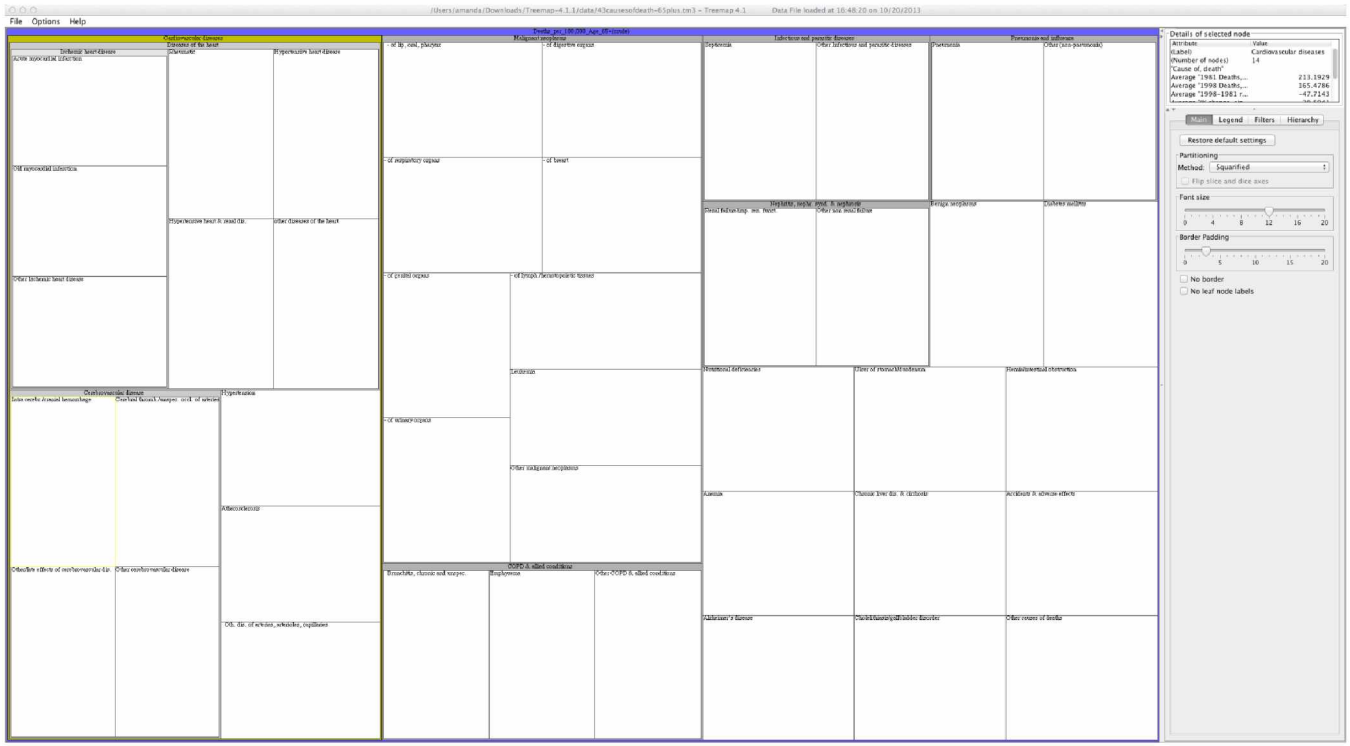
I switched off the "Leaders and Laggards" icon and then went to Edit -> Clear Queries to draw a new variable time box. Apparently once the "Leaders and Laggards" icon is activated even if you turn it off you will be unable to adjust the variable time box. I created a new variable time box in a different place on the graph and was able to see "Applied Micro Circuits" is a laggard, among others.



Appendix H - Tool Example – Treemap

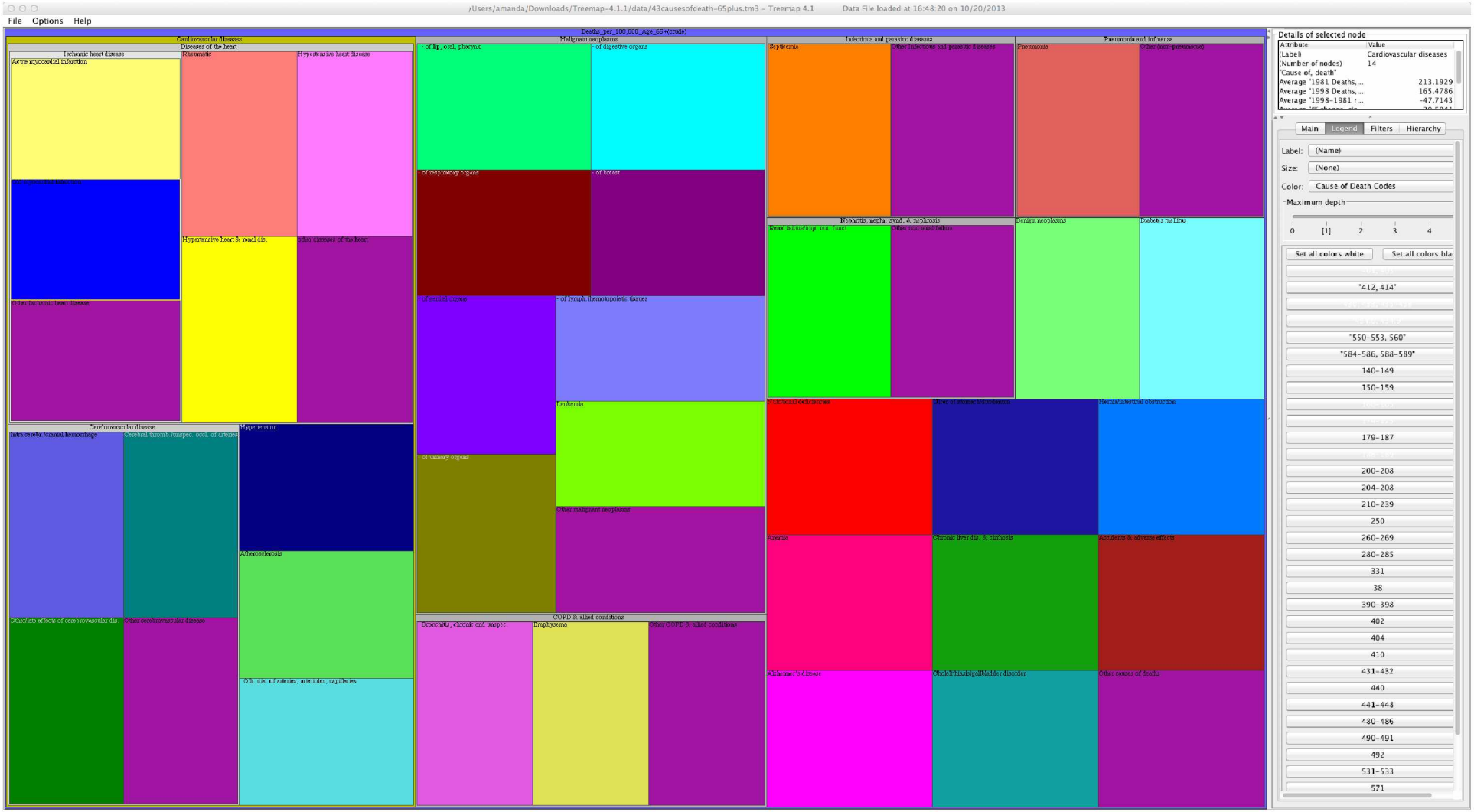
Treemap is a visualization tool available at http://www.cs.umd.edu/hcil/treemap/

Data: I used the sample data contained within the application for Causes of Death in 1981 and 1998 for ages 65 and up. On opening the application you can see the hierarchy is delineated by containers. (E.g., Cardiovascular diseases –> Diseases of the heart –> Ischemic heart disease)



150

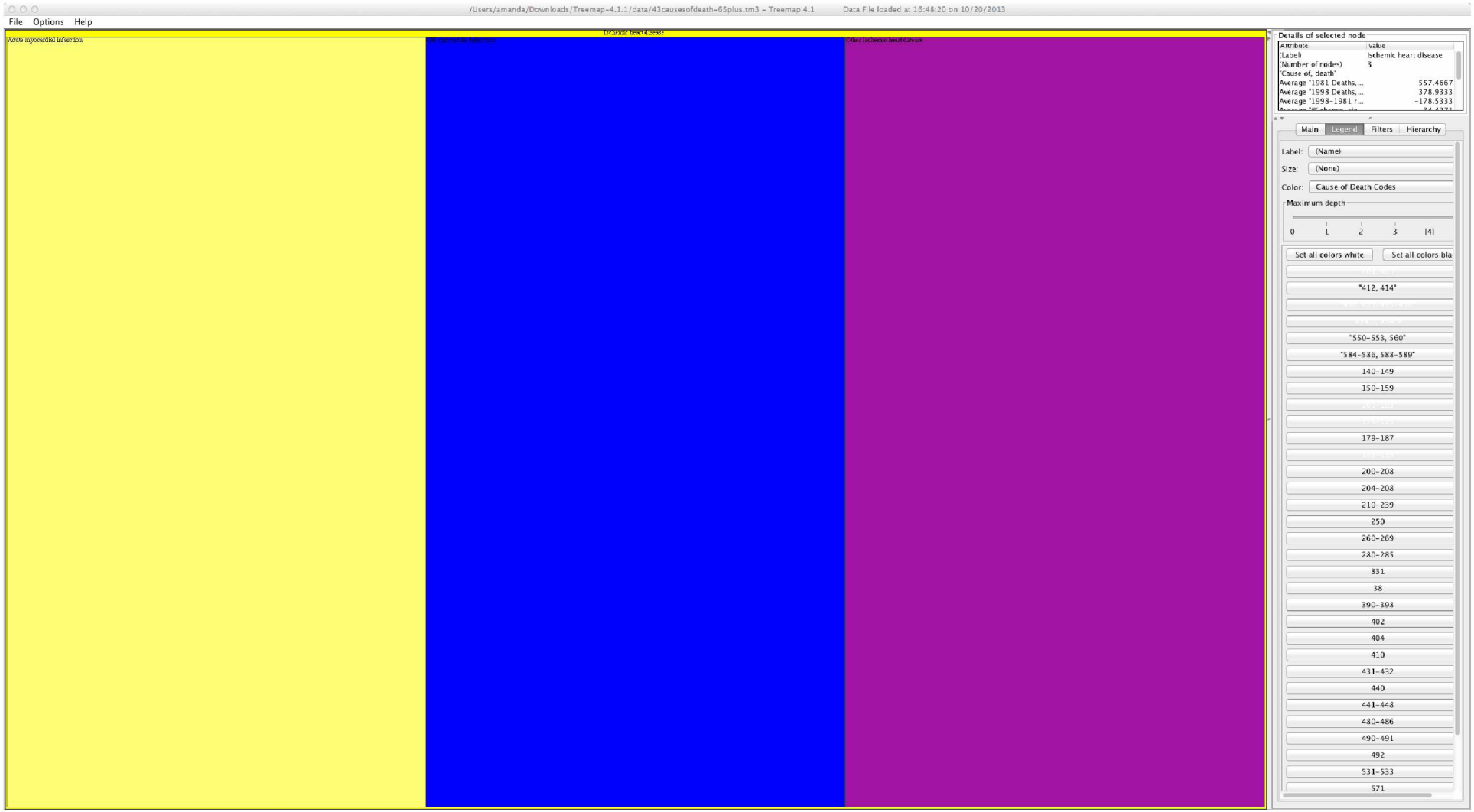
To improve the data visualization I turned on the colors by going to the Legend -> Color drop down and selecting "Cause of Death Codes".



I then drilled down into the hierarchy of the Cardiovascular disease map by double clicking on the header.

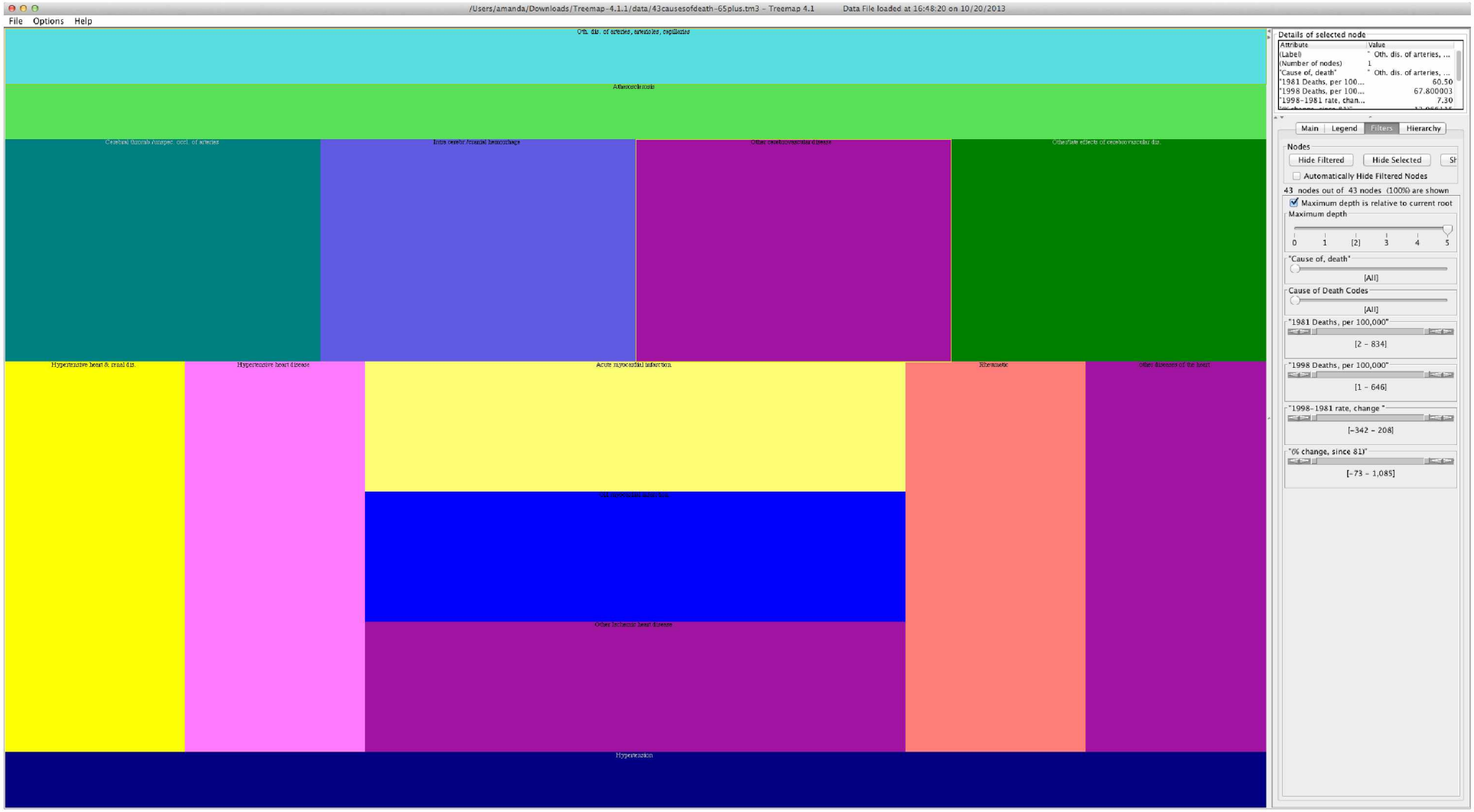
152

I went further down by double clicking the Ischemic heart disease header.



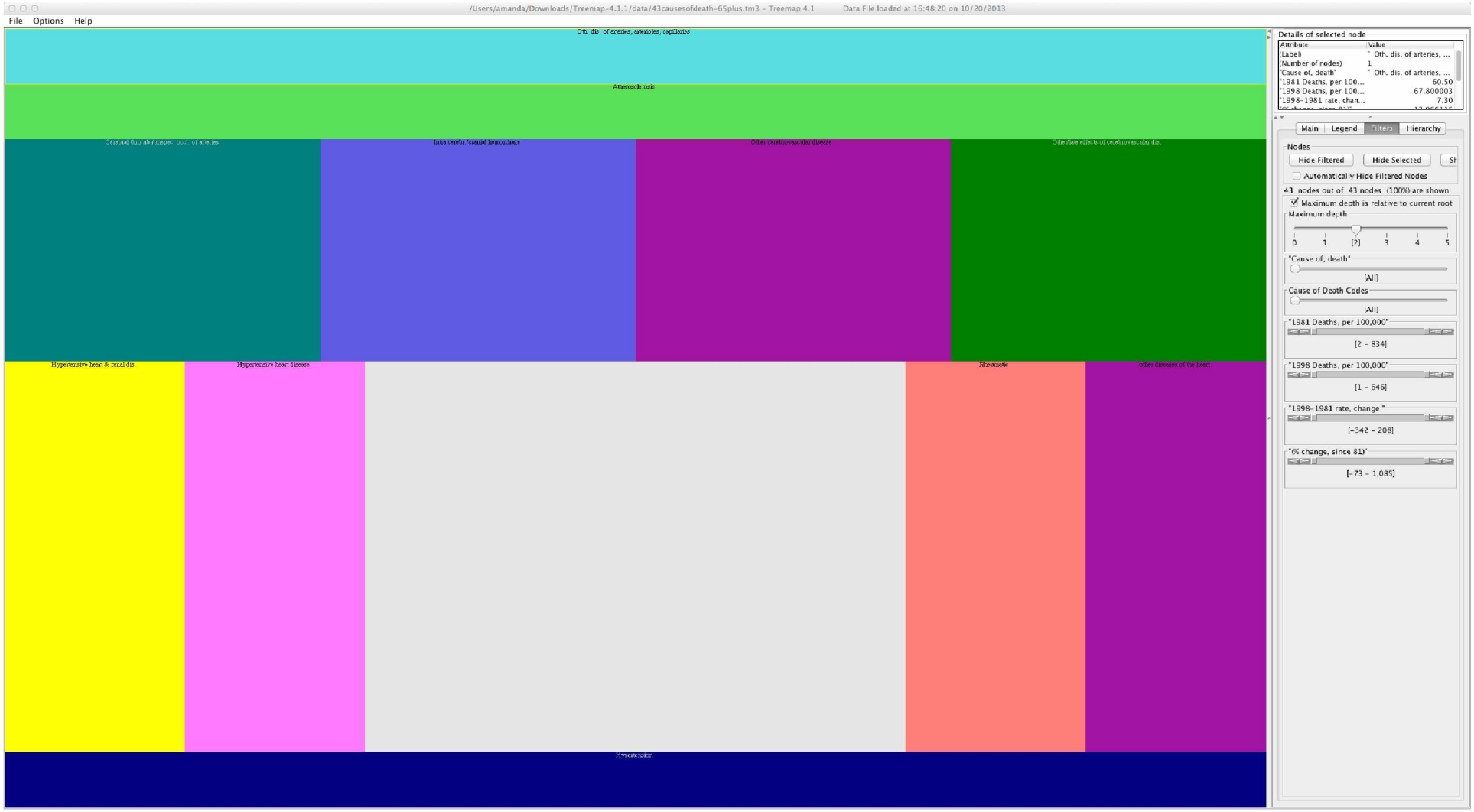
I returned to Cardiovascular disease by pressing Esc twice. Then I changed the partitioning method by going to Main -> Partitioning Method and selecting Strip.

I again experimented with the partitioning method by changing that option to “Slice and Dice”. I turned off borders by checking the “No border” checkbox.



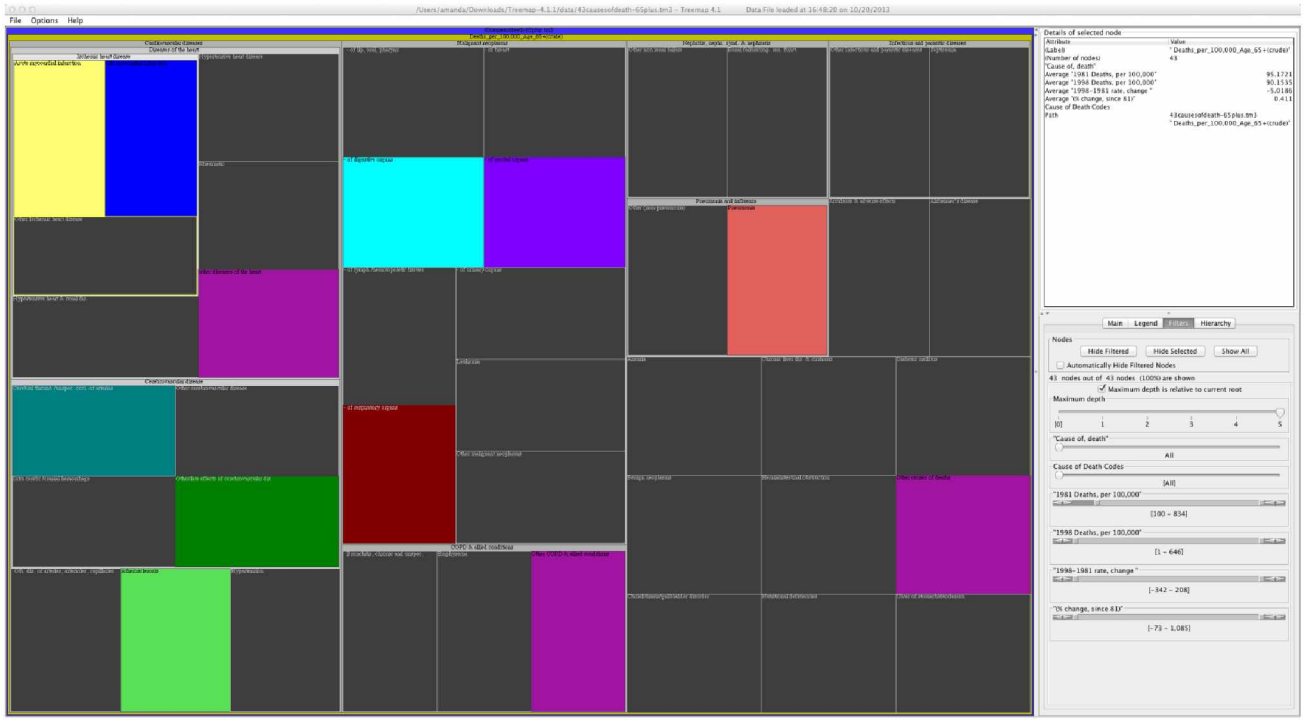
I switched to the Filters tab.

I adjusted the maximum node depth by setting the slider to 2. Note that the Ischemic heart disease nodes drop off the map.

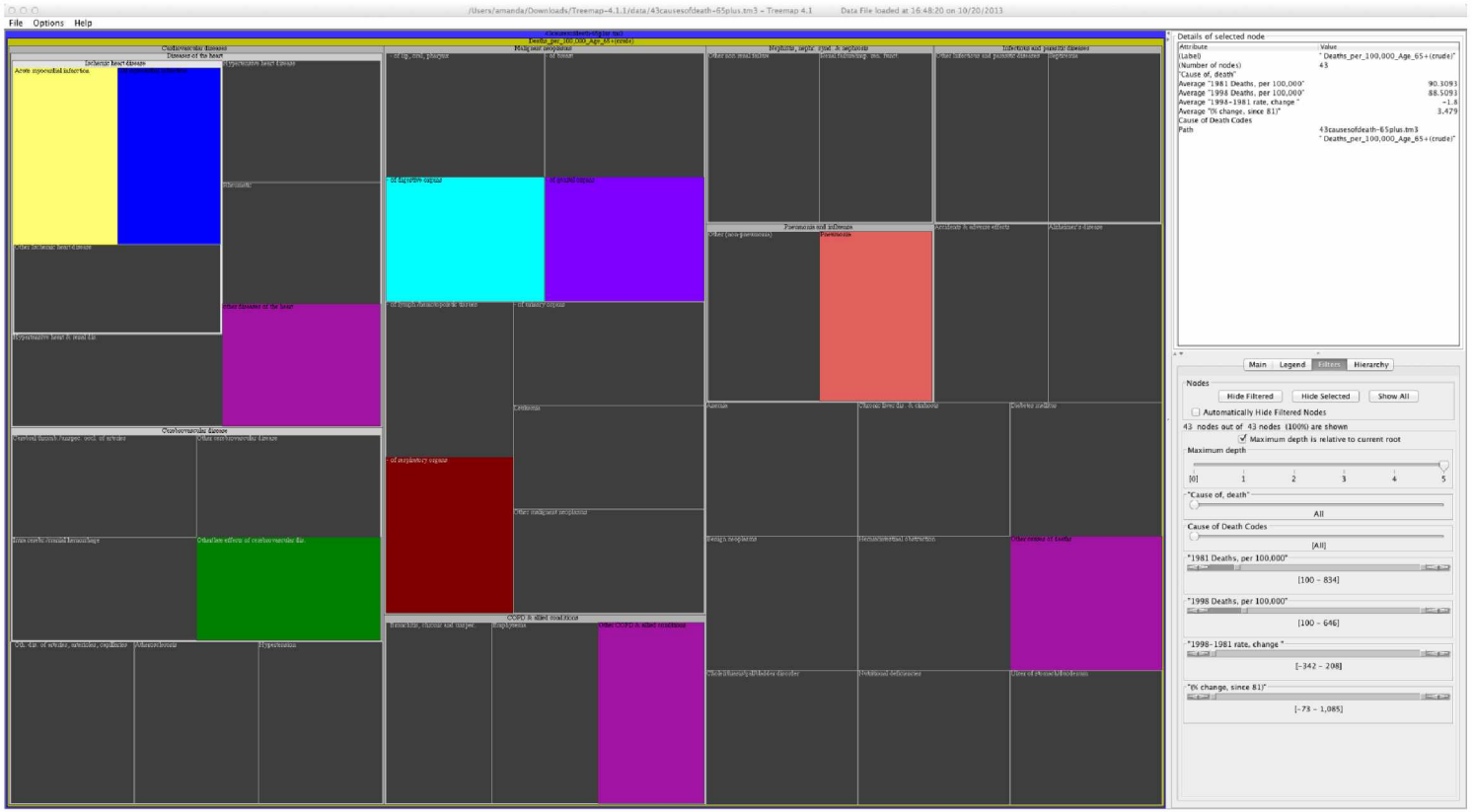


I reset the maximum node depth to 5. I hit Esc twice to return to the full tree.

I switched to the Legend tab and set the partitioning method to Squarified and unchecked the "No borders" box. I then switched back to the Filter tab where I adjusted the slider for "1981 Deaths, per 100,000" to a minimum value of 100 so I could see what diseases caused a minimum of 100 deaths for every 100,000 in 1981 and a minimum of 1 death for every 100,000 in 1998. Note that Pneumonia, Acute myocardial infarction, etc., are the only nodes still colored.



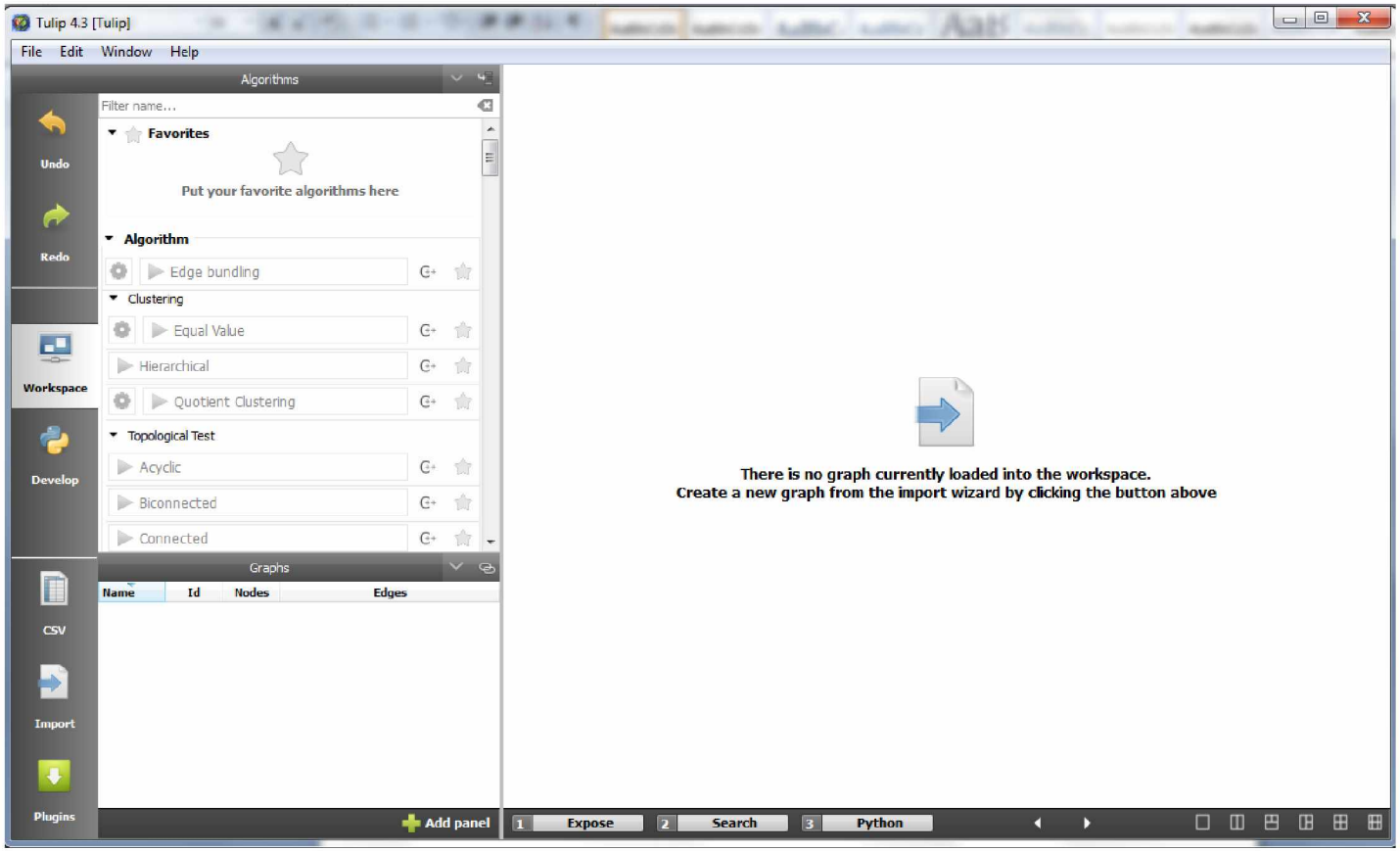
I then set the slider for "1998 Deaths, per 100,000" to a minimum value of 100. This shows nodes where the minimum deaths in both 1981 and 1998 were at least 100 for every 100,000 people. Note that two Cerebrovascular diseases drop off the map. If the slider for 1981 were set back to 2 (its minimum possible value) then you would see that Diabetes mellitus suddenly appears as it was a slightly greater cause of death in 1998 than in 1981. If the slider for 1981 is adjusted to 97 then Diabetes mellitus reappears.



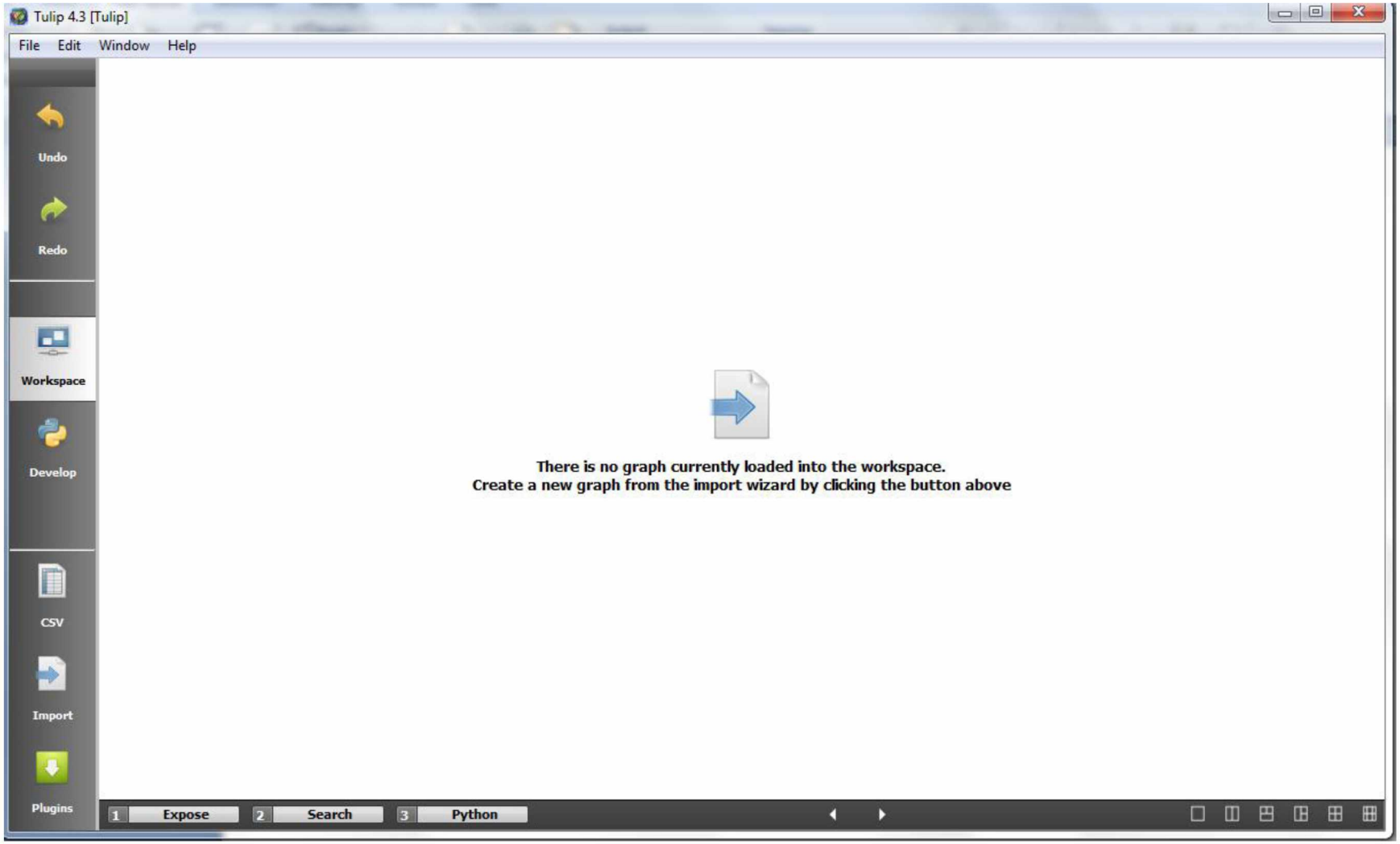
Appendix I - Tool Example – Tulip

Tulip is a linked graph visualization tool available at http://tulip.labri.fr/TulipDrupal/

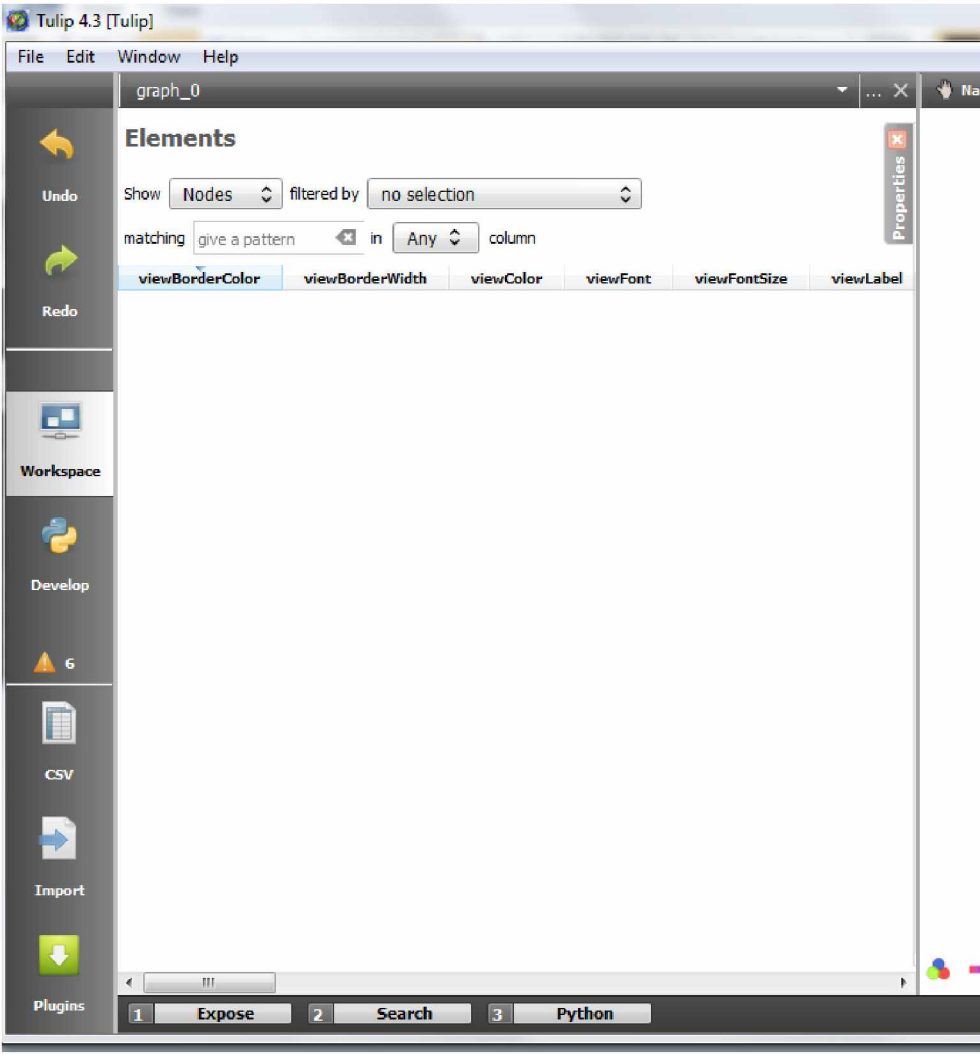
I ran Tulip 4.3 and was shown the home screen.

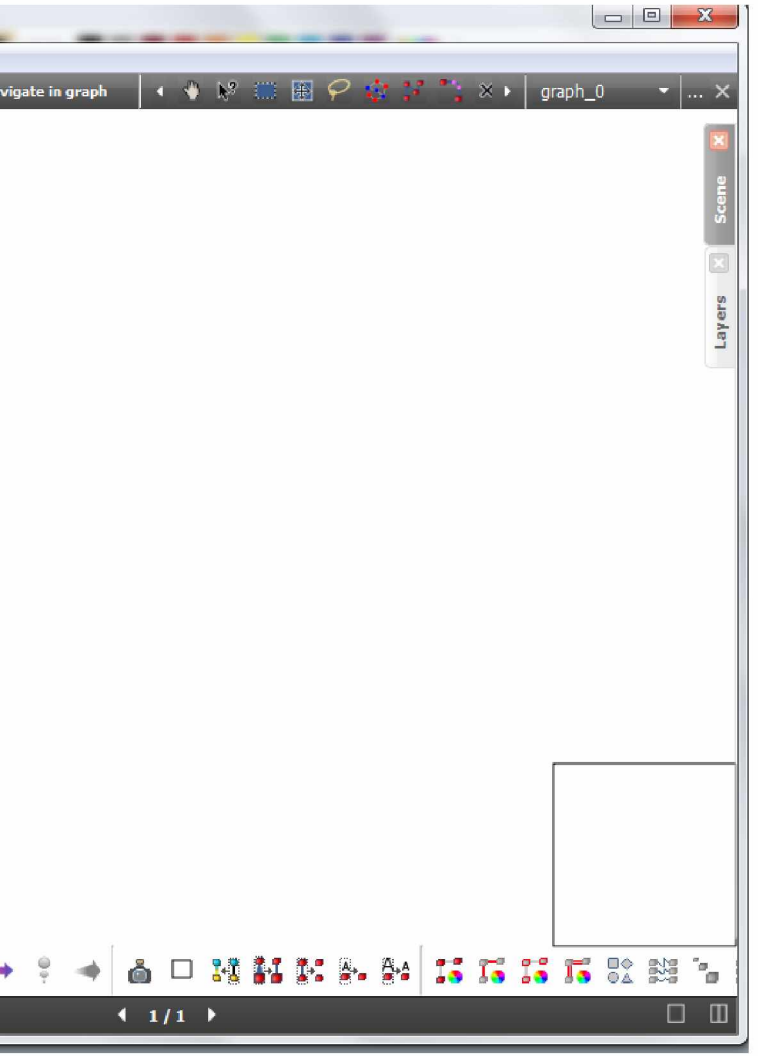


Hid the Algorithms window by resizing the window to hide it.



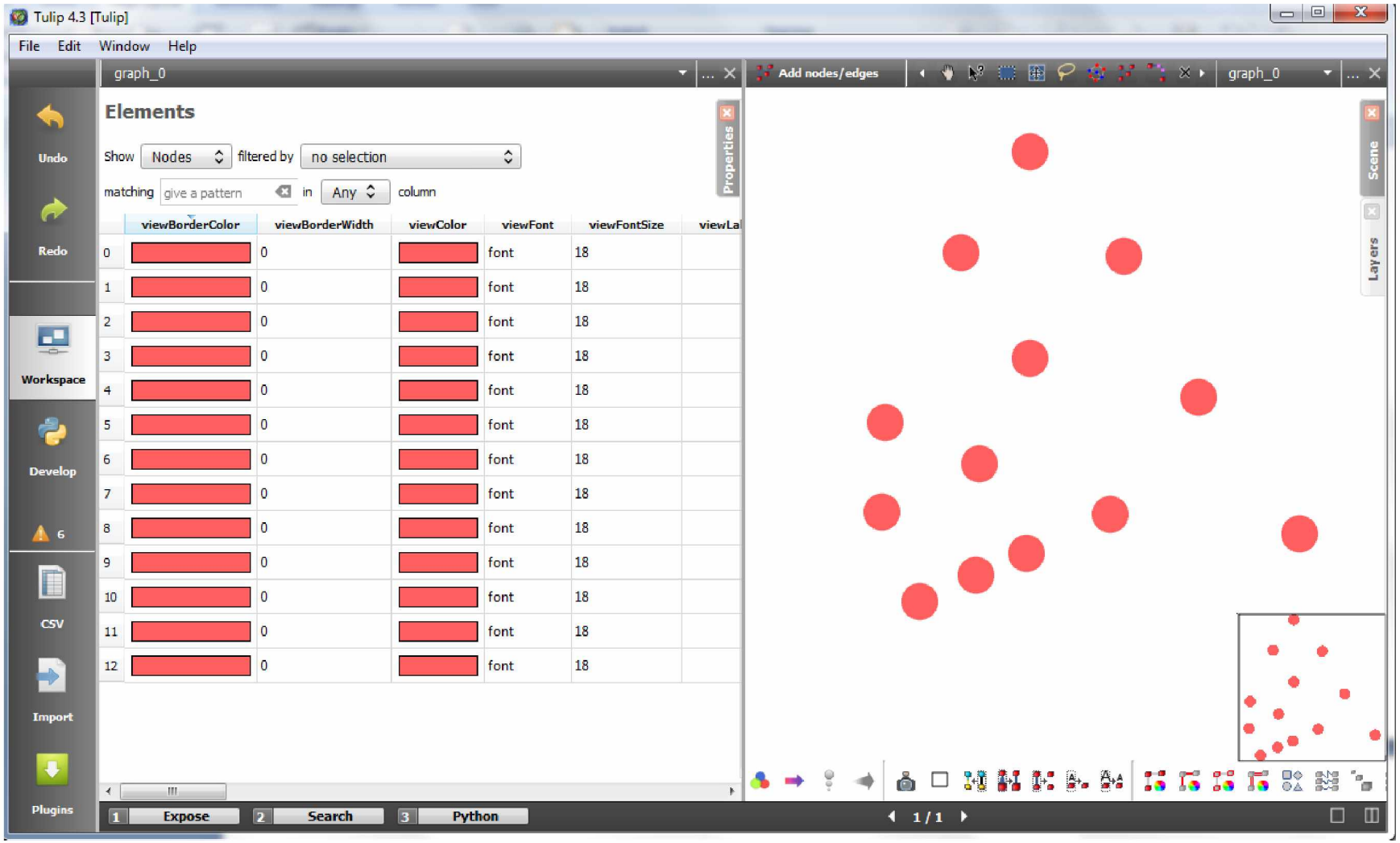
Went to File -> New Graph.



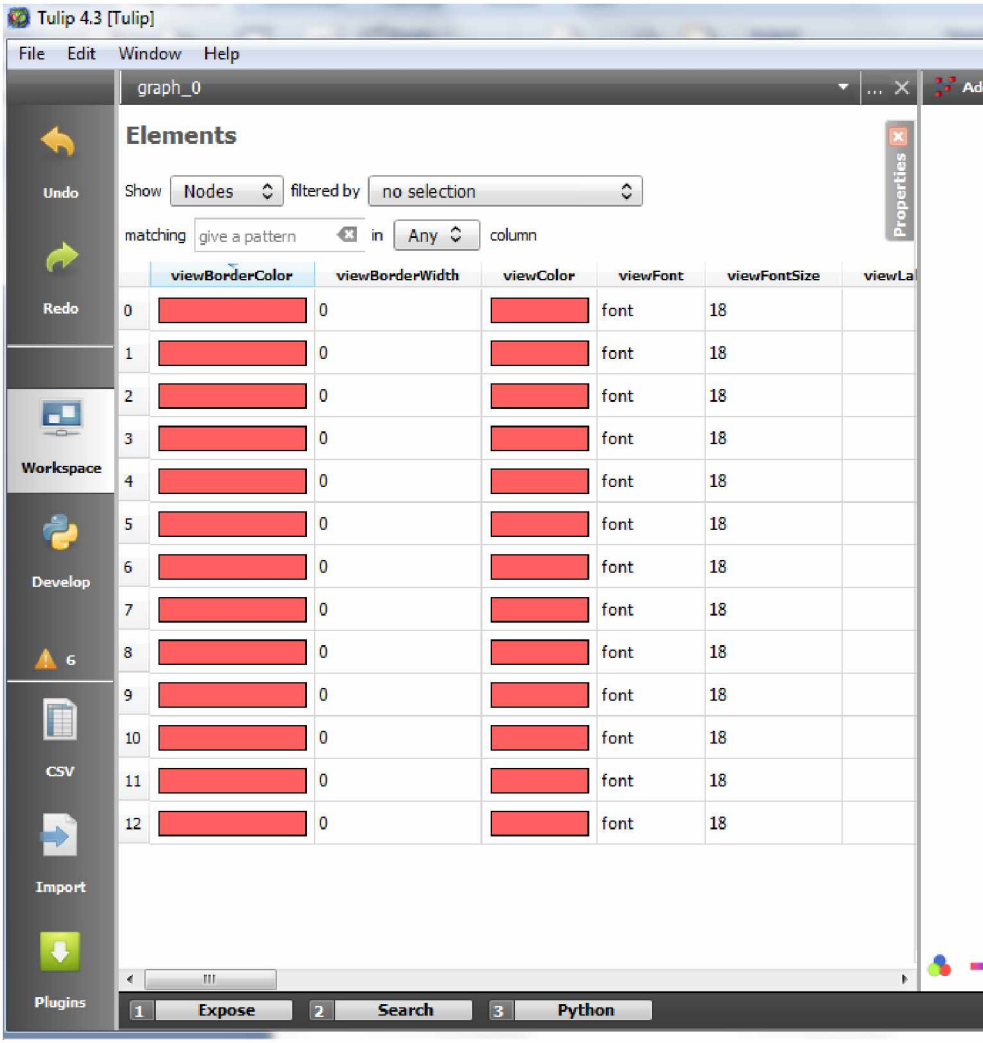

vigate in graph
graph_0
Scene
Layers
1 / 1

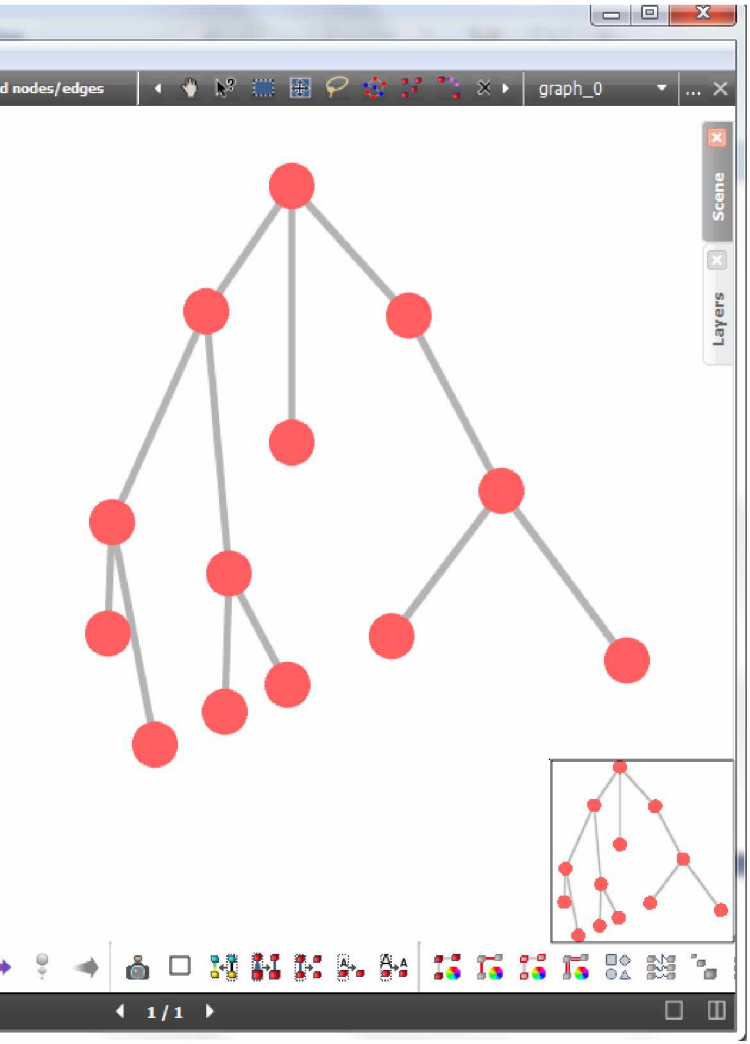
Selected the “Add nodes/edges” tool from the upper toolbar -> 

Placed a few nodes in the drawing window.

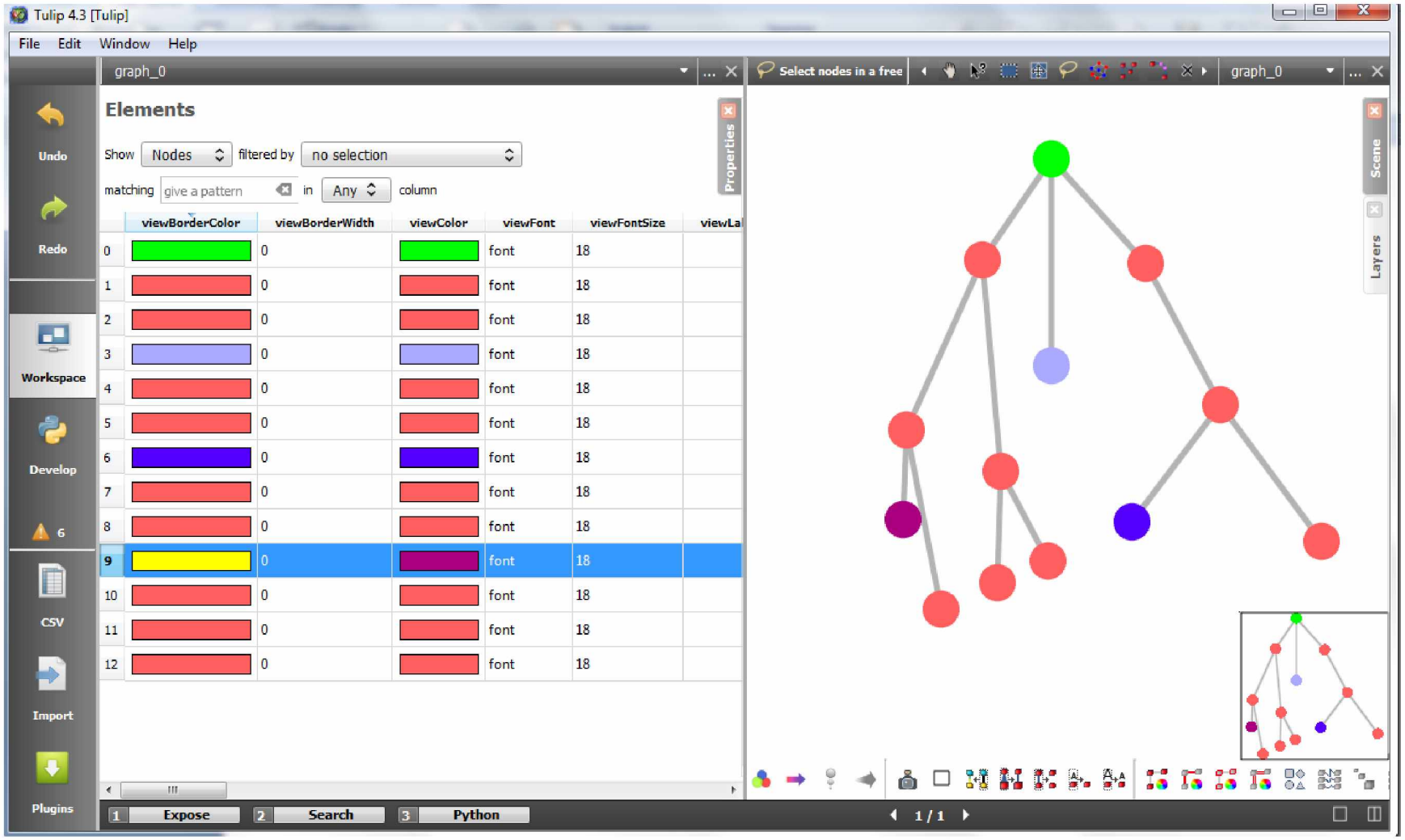


Drew edges between the nodes.



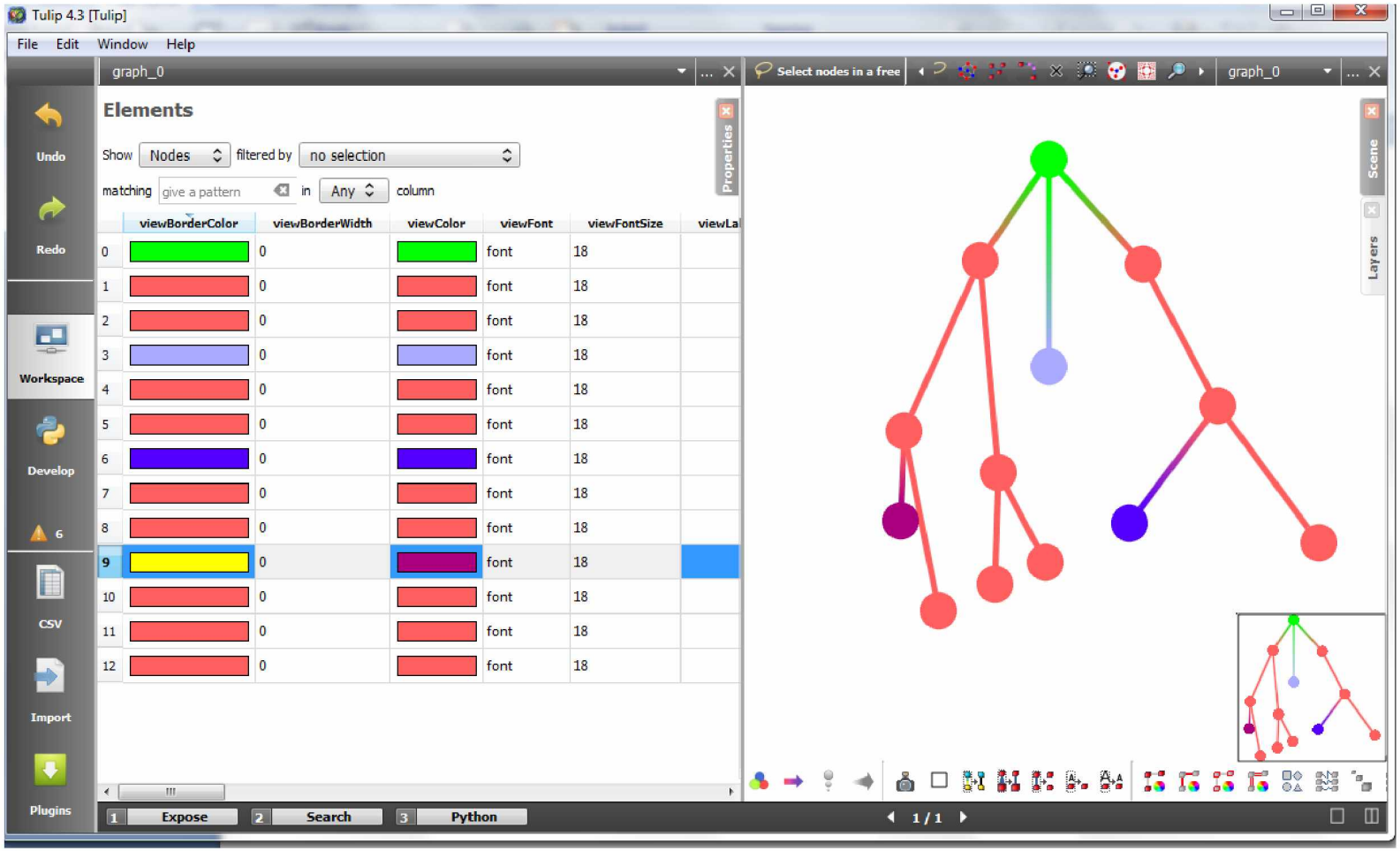

d nodes/edges
graph_0
Scene
Layers
1 / 1

Changed the view color and view border color on a few of the nodes in the table to the left. Note that the colors on the nodes in the linked graph automatically change.



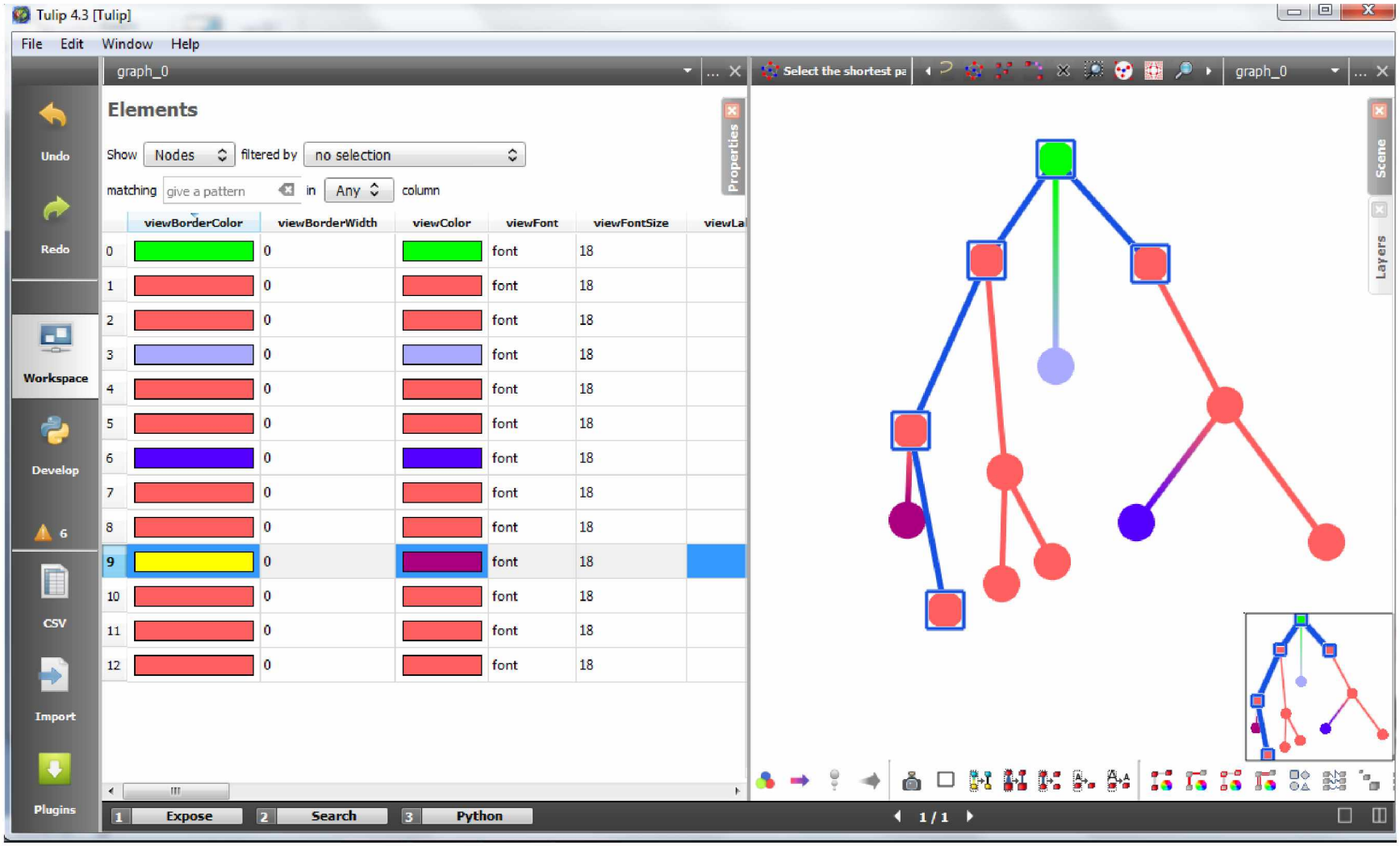
Selected to “Toggle edge color interpolation” -> 

Note how the edge colors are now gradients depending on the colors selected in the Elements table to the left.

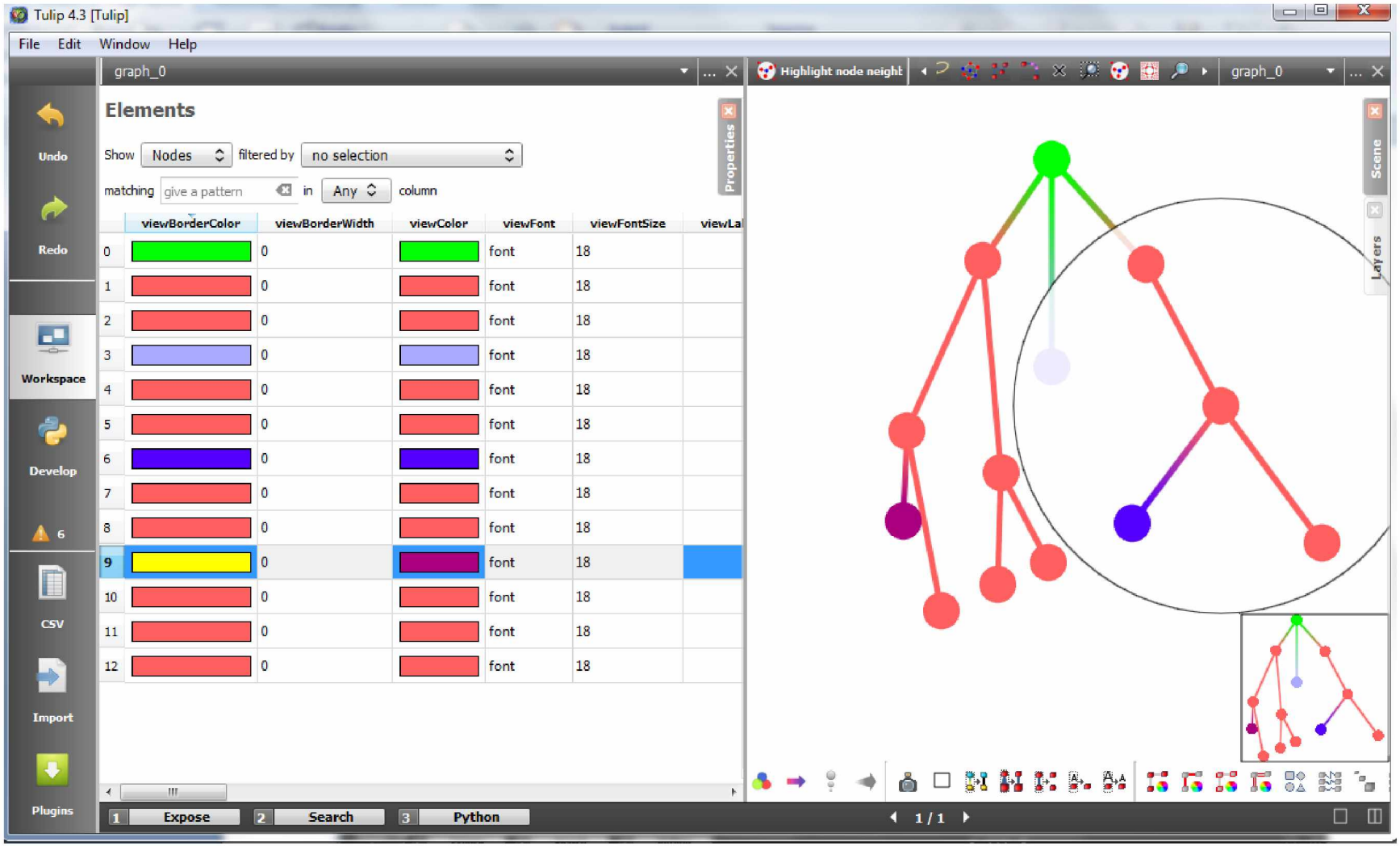


Used the “Select the shortest path between two nodes” tool -> 

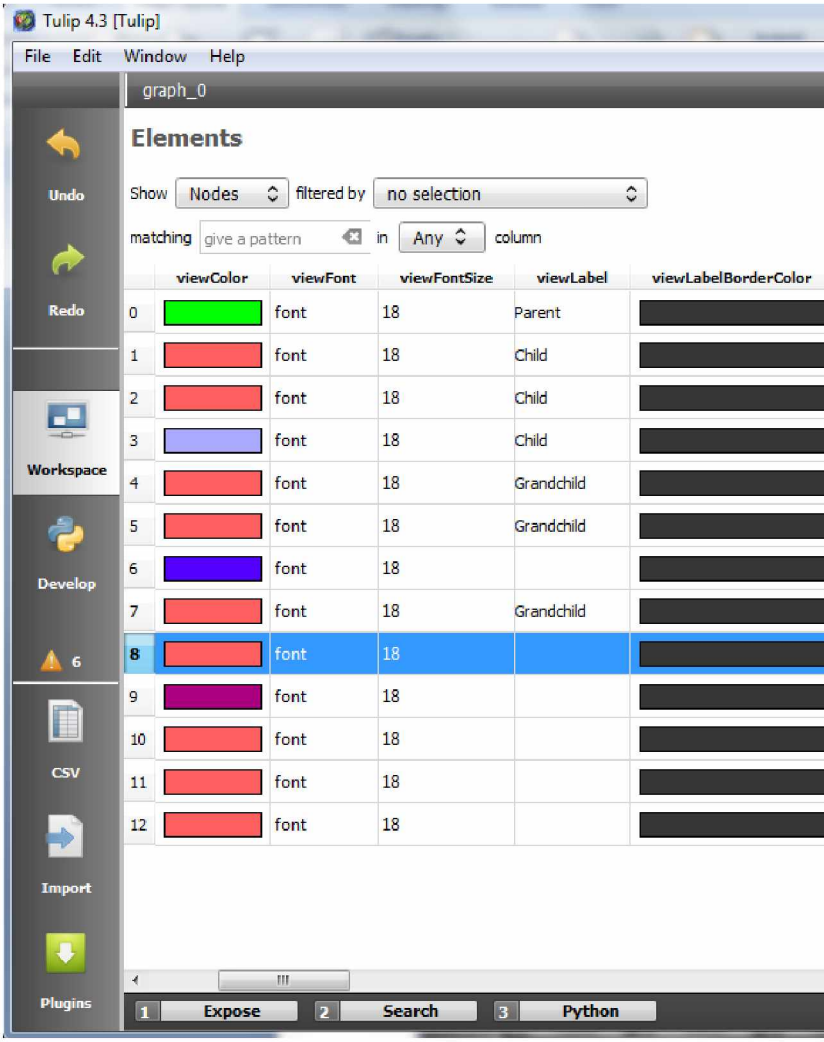
Note how the shortest path is automatically indicated in blue between the two selected nodes.



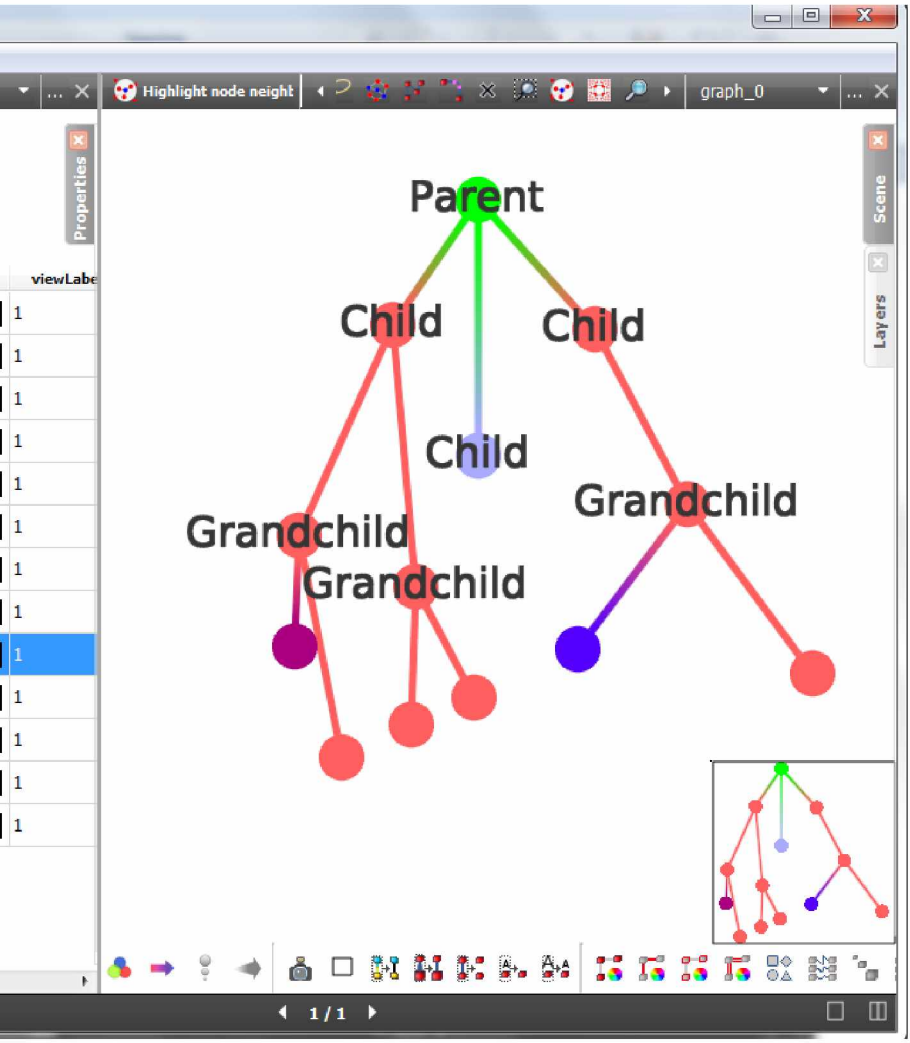
Selected “Highlight node neighbors” tool ->  .



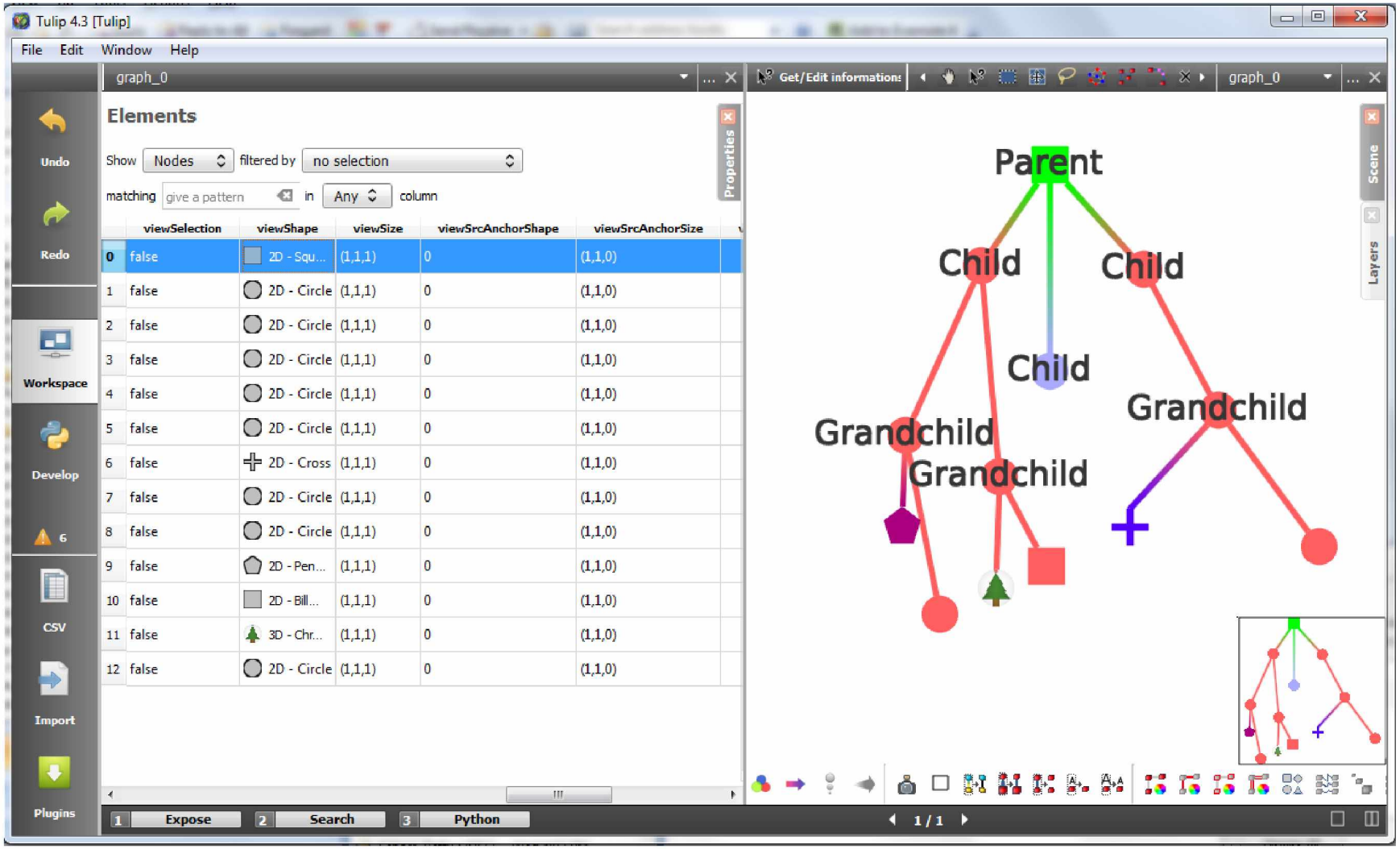
Updated the viewLabel field for some of the nodes;



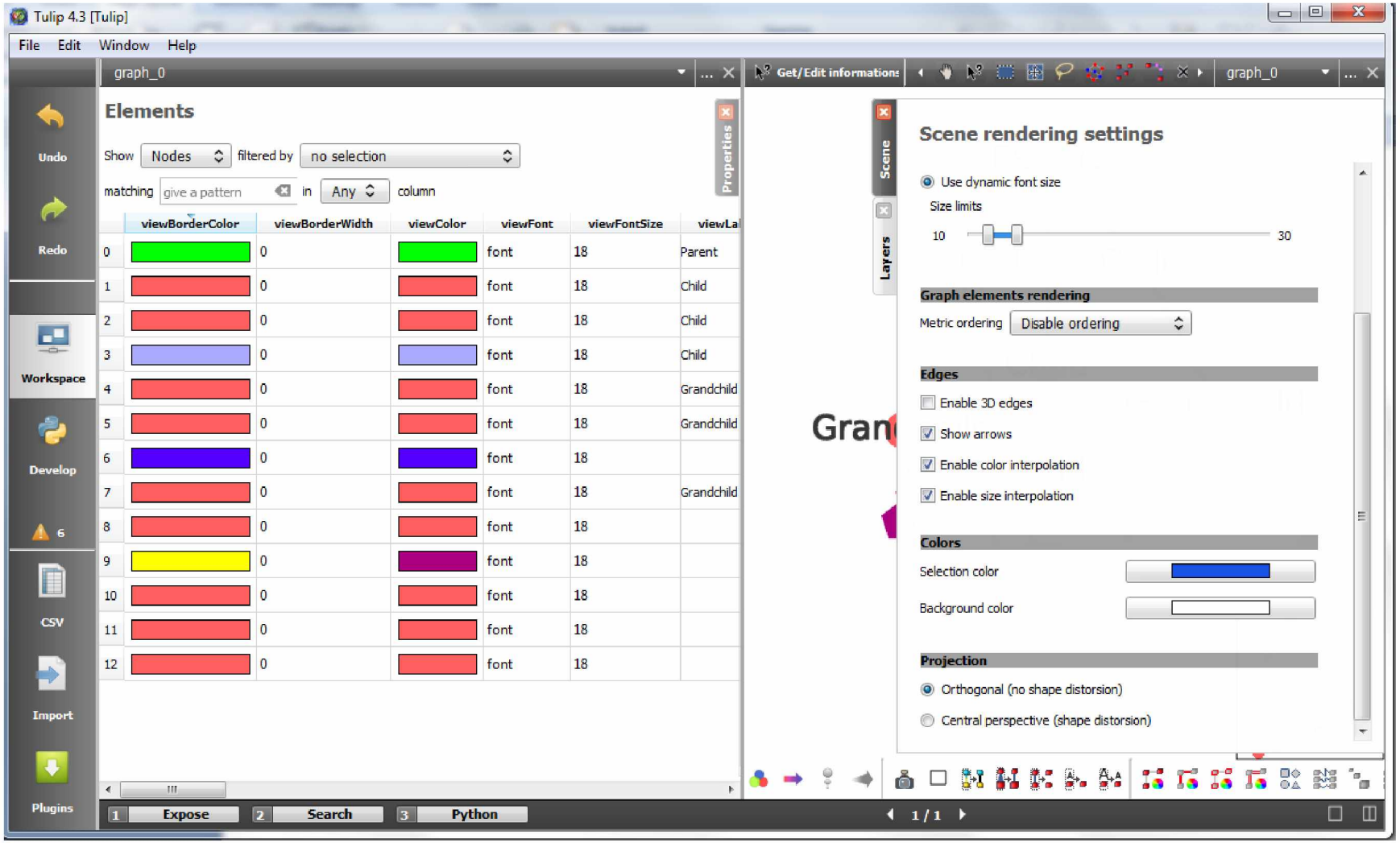
note how the linked graph updates as well.



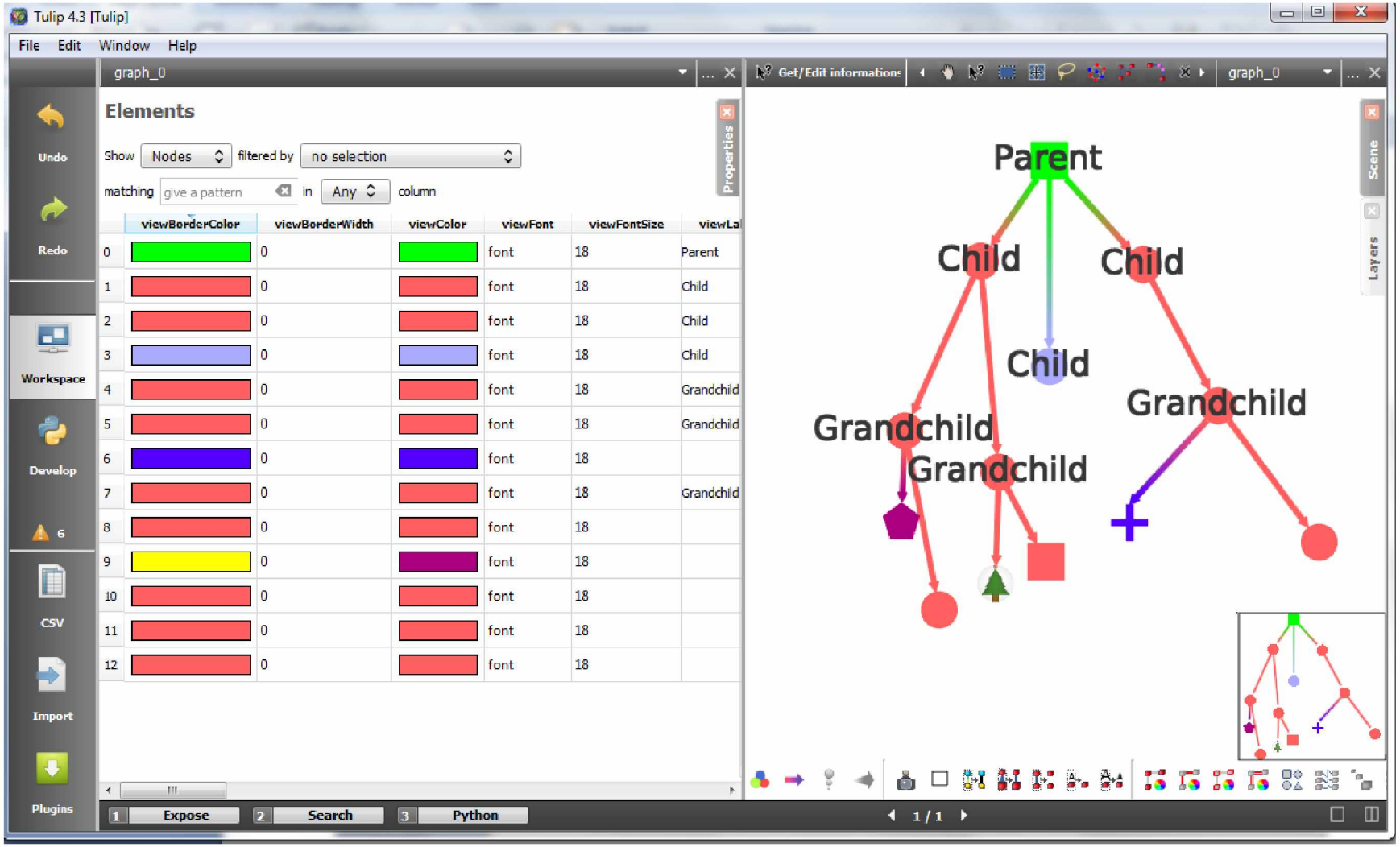
Updated the view shape for some of the nodes.



Selected the Scene menu on the right. Checked “Show Arrows” box.



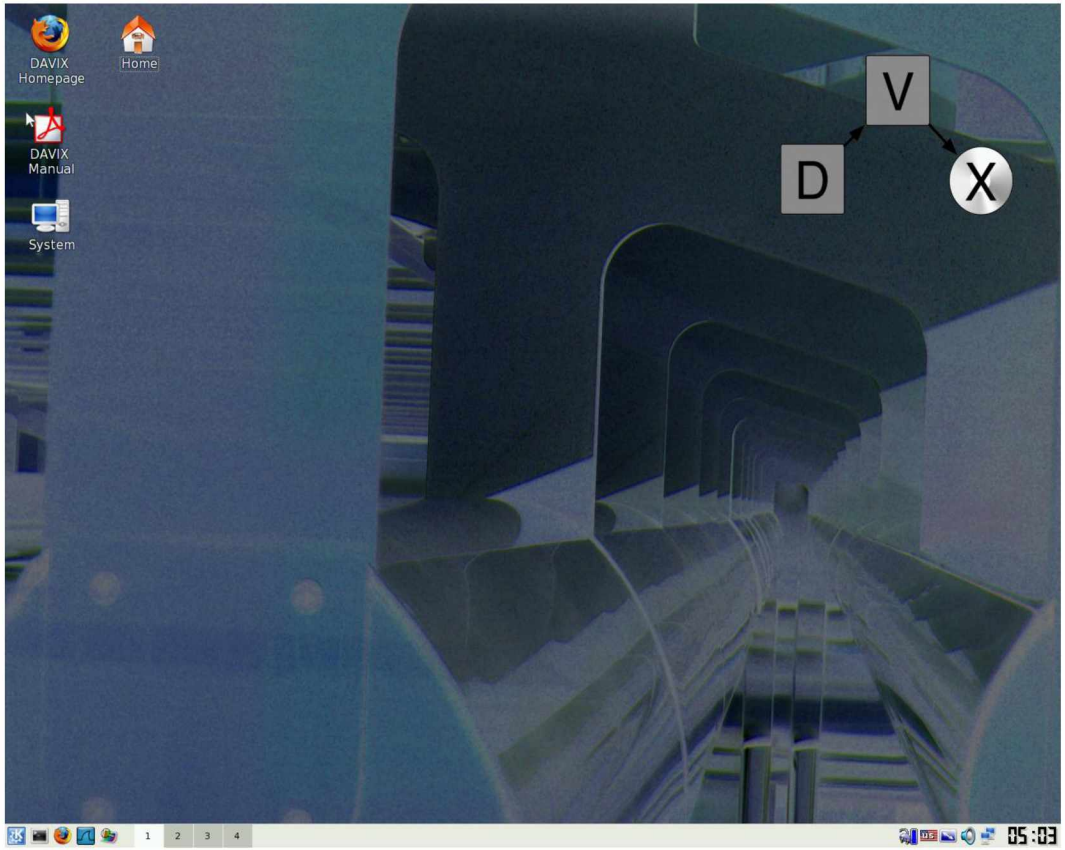
Hid the Scene menu. Note arrows are now displayed in the graph.



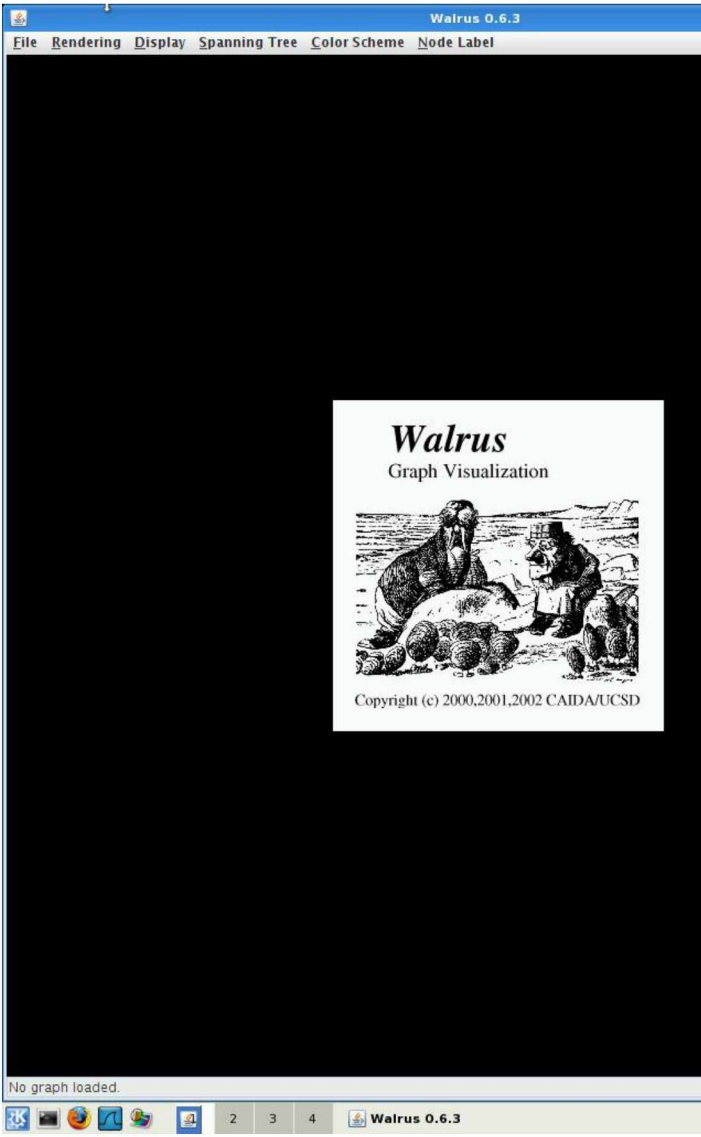
Appendix J - Tool Example – Walrus

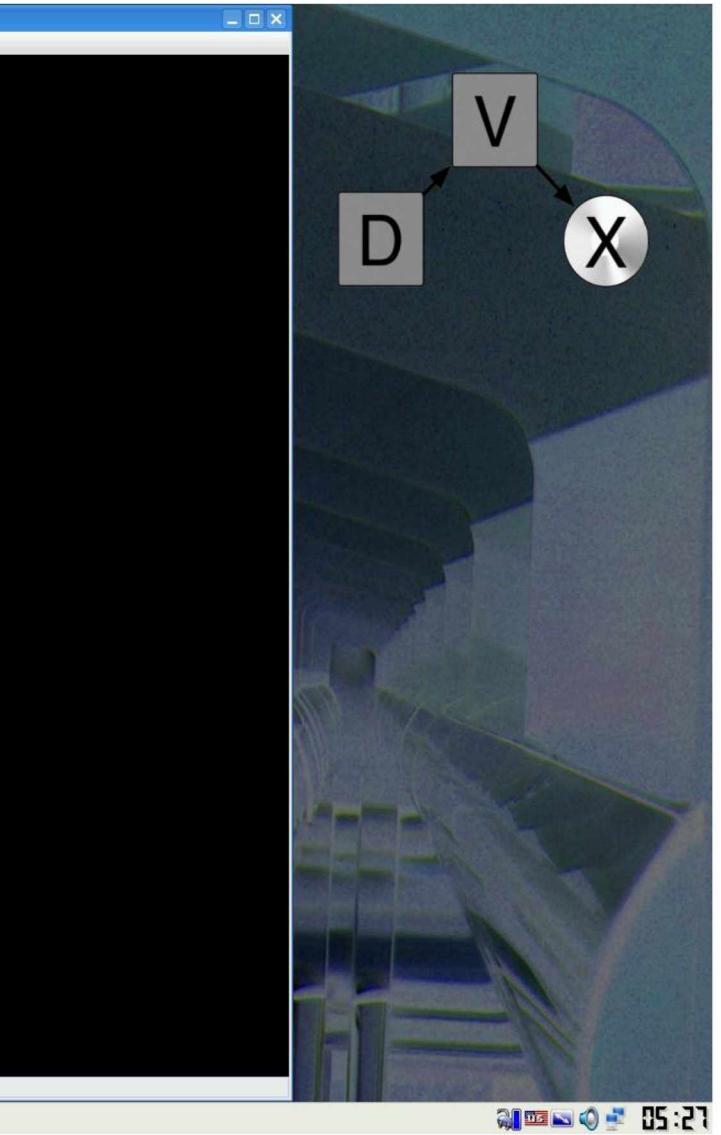
Walrus which allows visualization of hierarchical data as 3D linked graphs. It is available at http://www.caida.org/tools/visualization/walrus/

I ran Walrus from DAVIX and thus began at the DAVIX home screen:

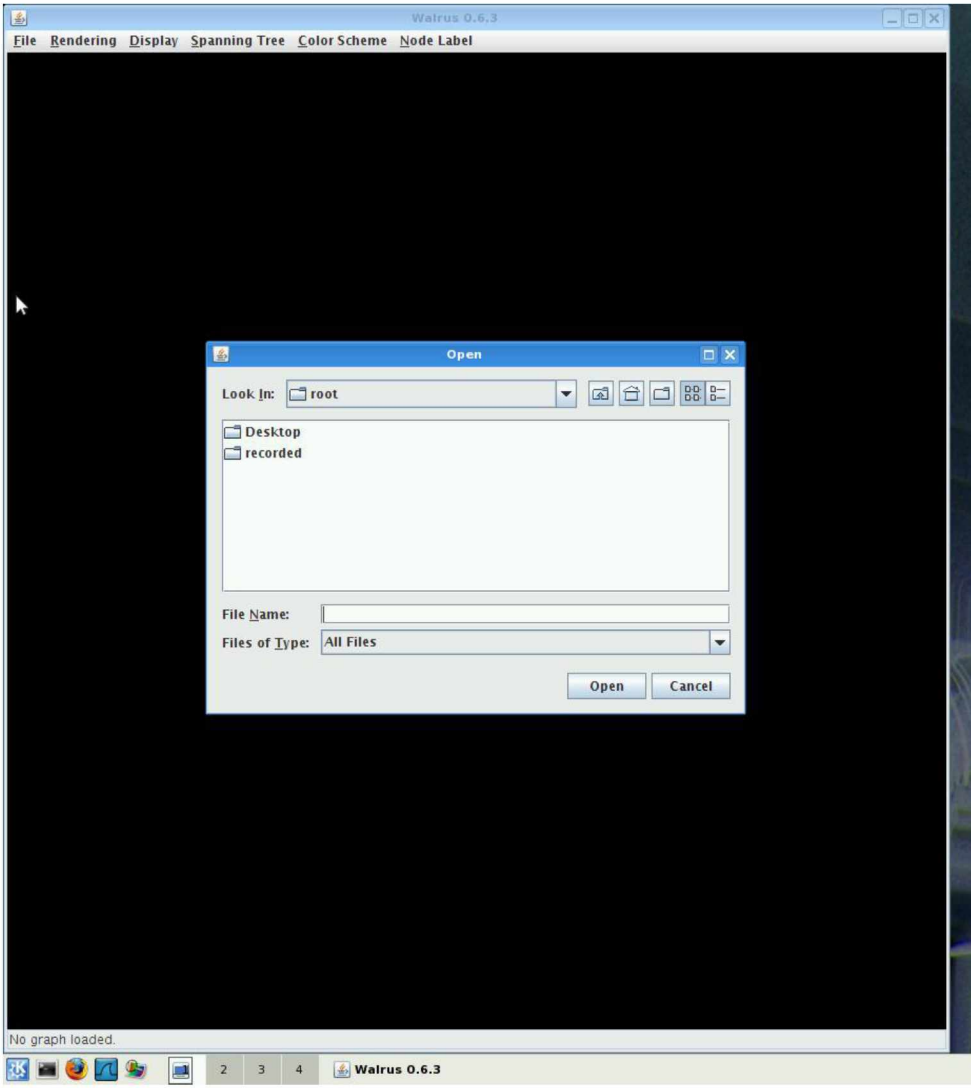


Then I opened Walrus.




V
D
X
05:27

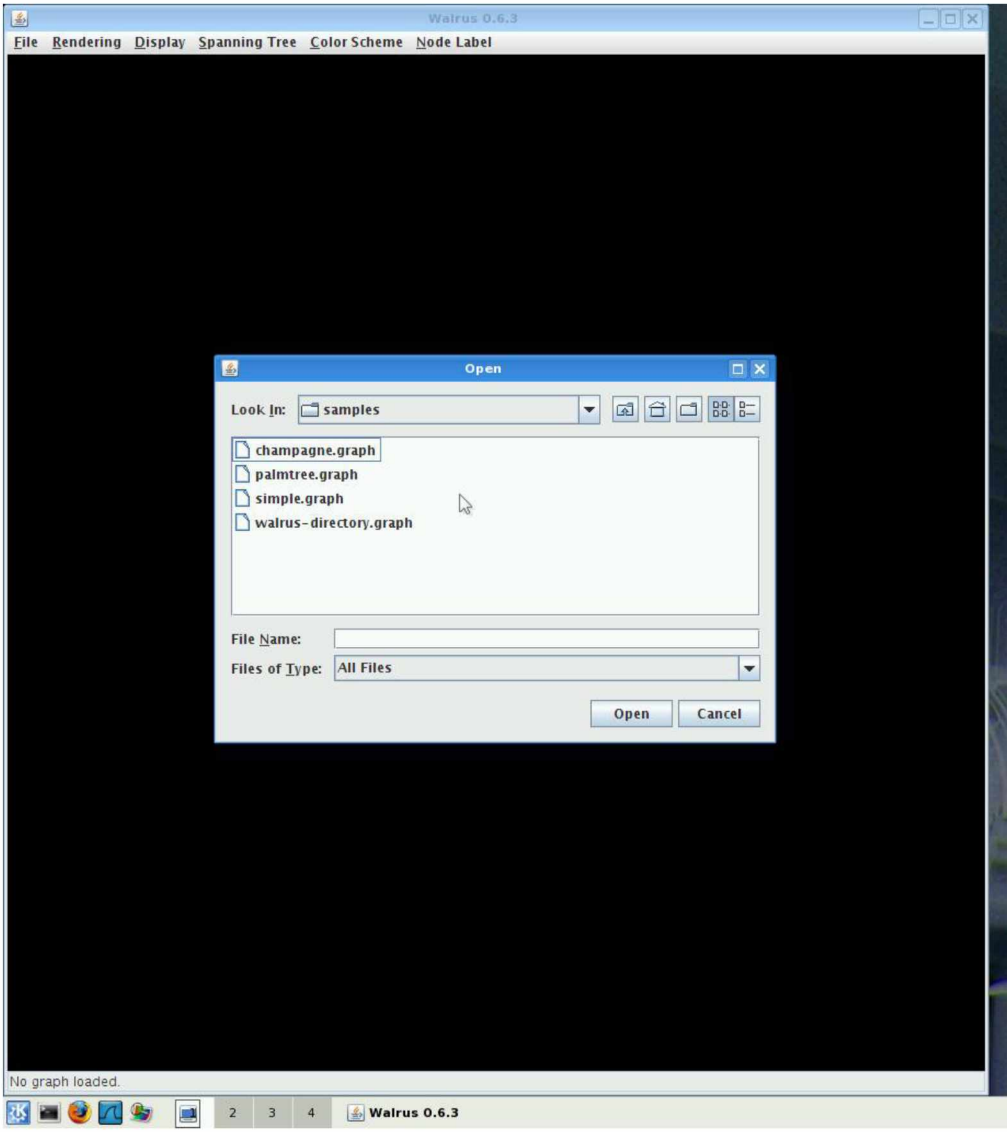
I went to File -> Open




V
D
X
05:29

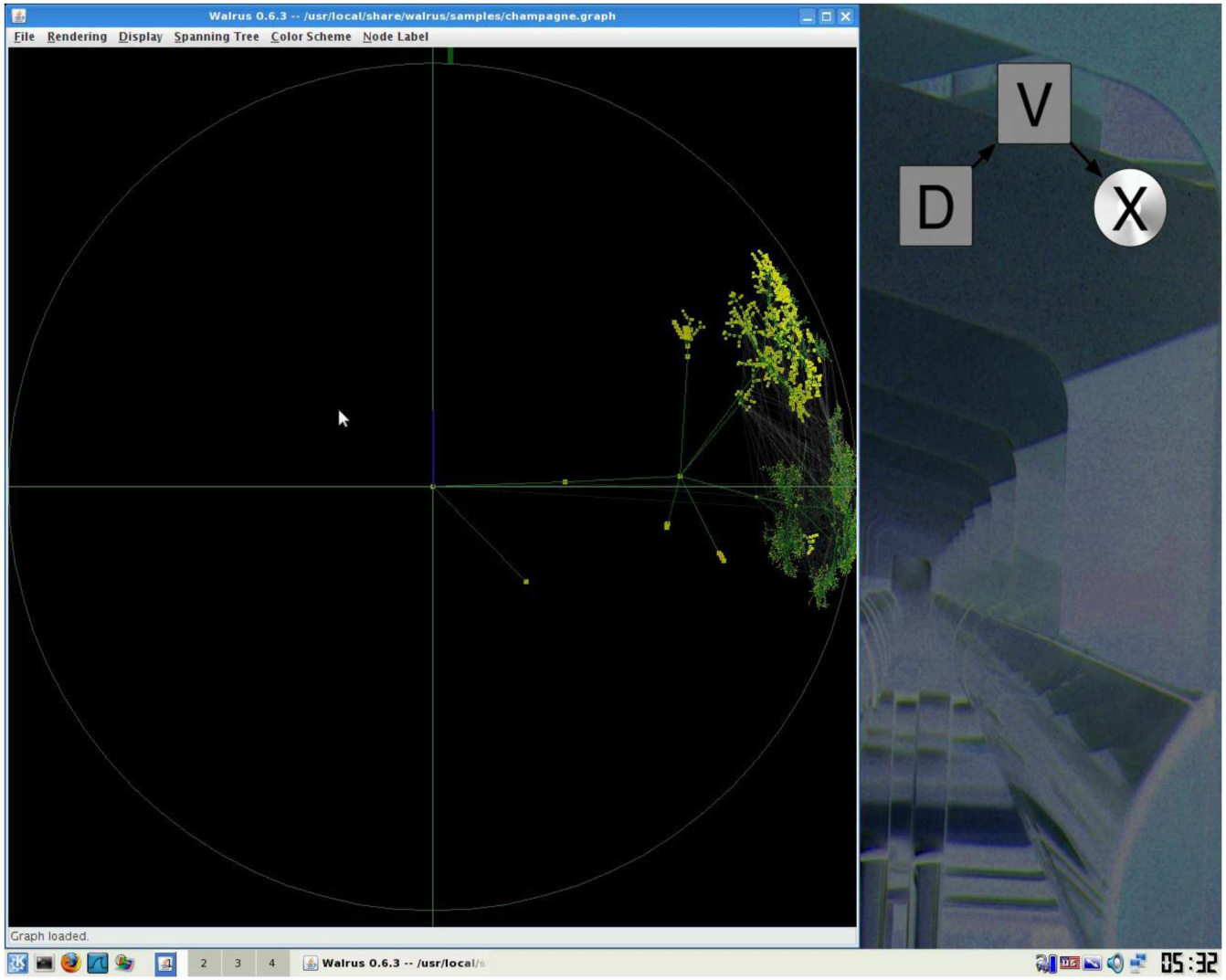
176

I then navigated to /usr/local/share/walrus/samples

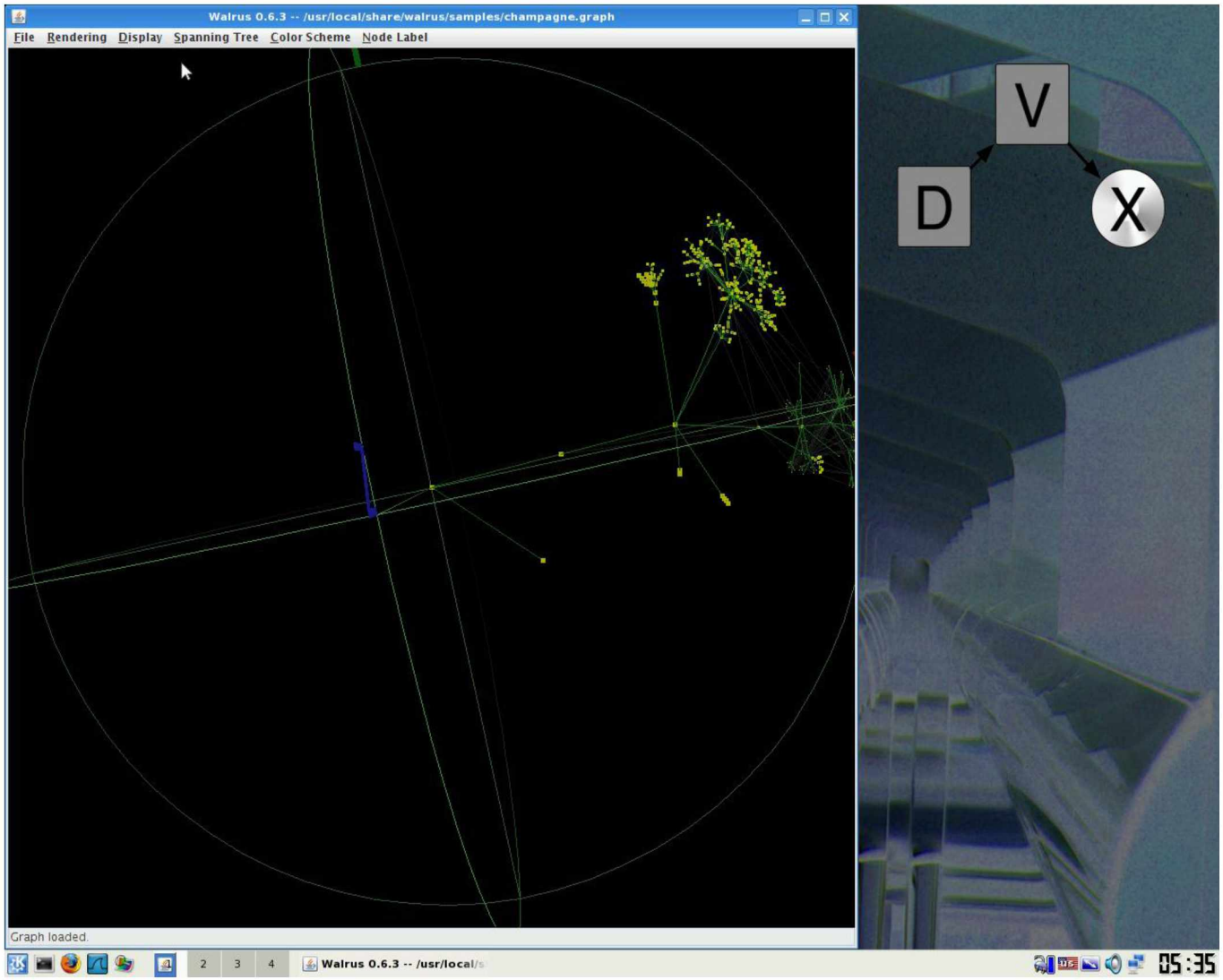



V
D
X

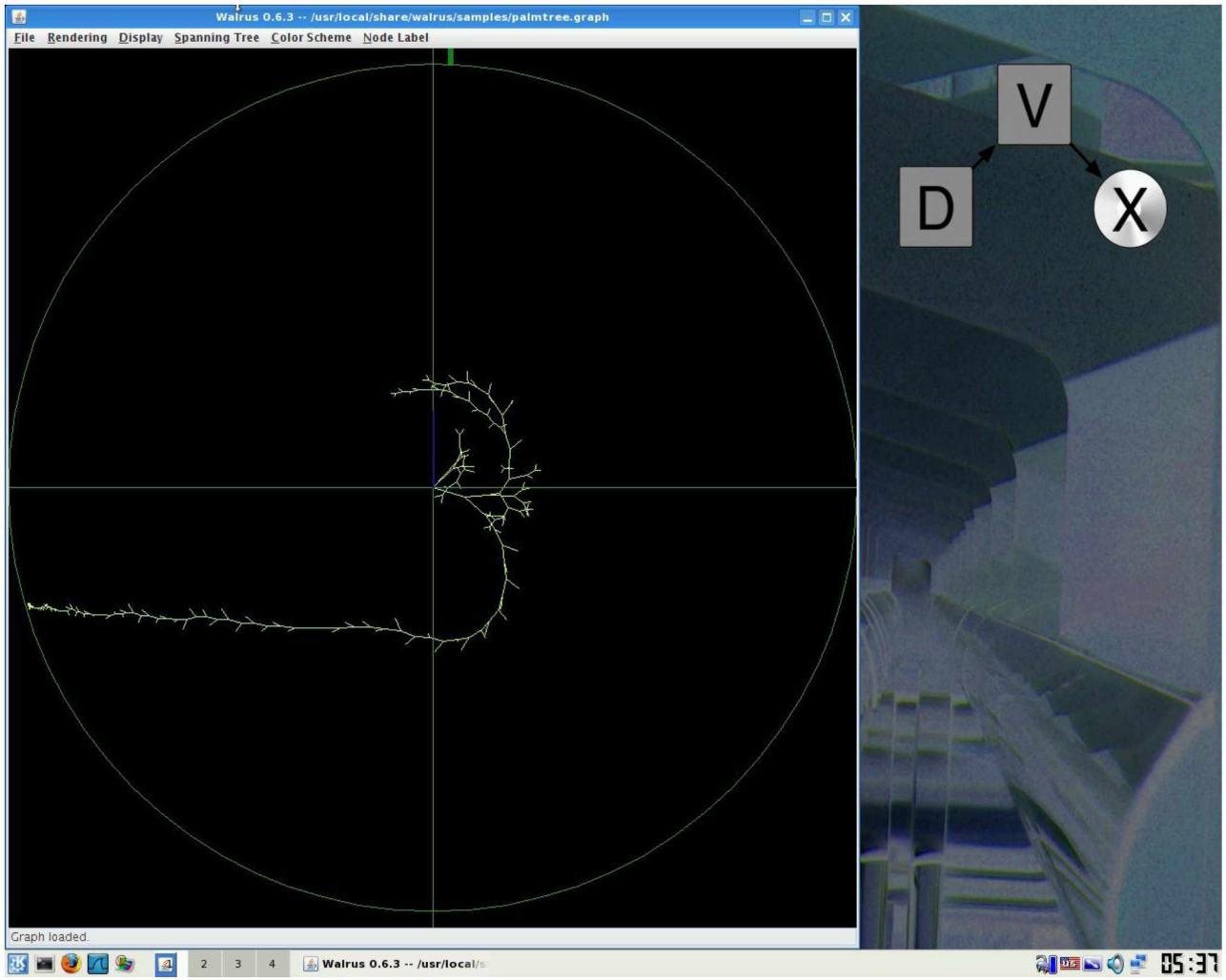
I selected champagne.graph and opened it. I then went to Rendering -> Start. A graph displayed.



I went to Rendering -> Wobble and the orb began wobbling. I then went to Rendering -> Prune Neighbor -> Distance <= 5. Note this image has rotated from the previous one and there are fewer nodes displayed.

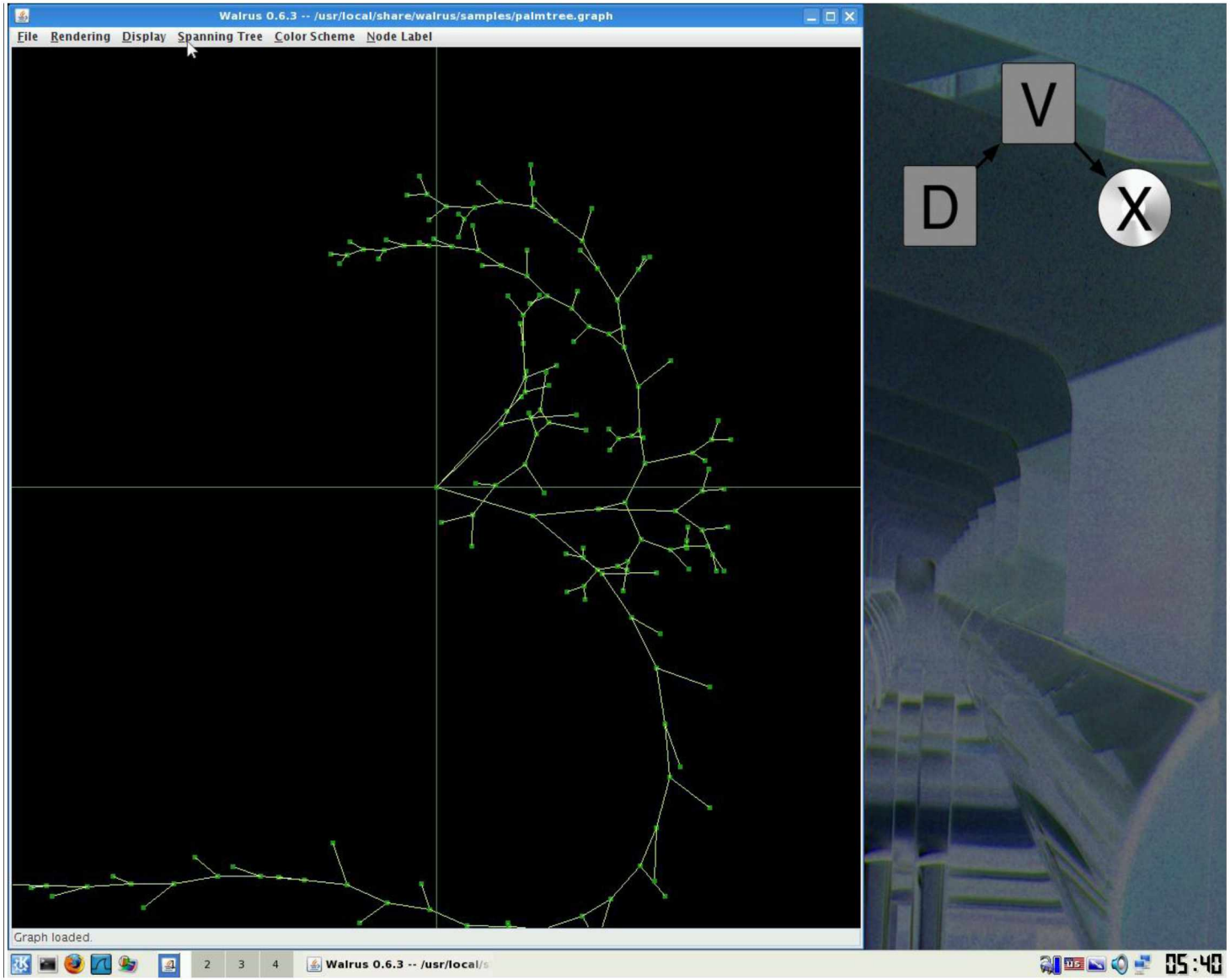


I went back to File -> Open and selected palmtree.graph and set Rendering -> Start. This is a notably different image.

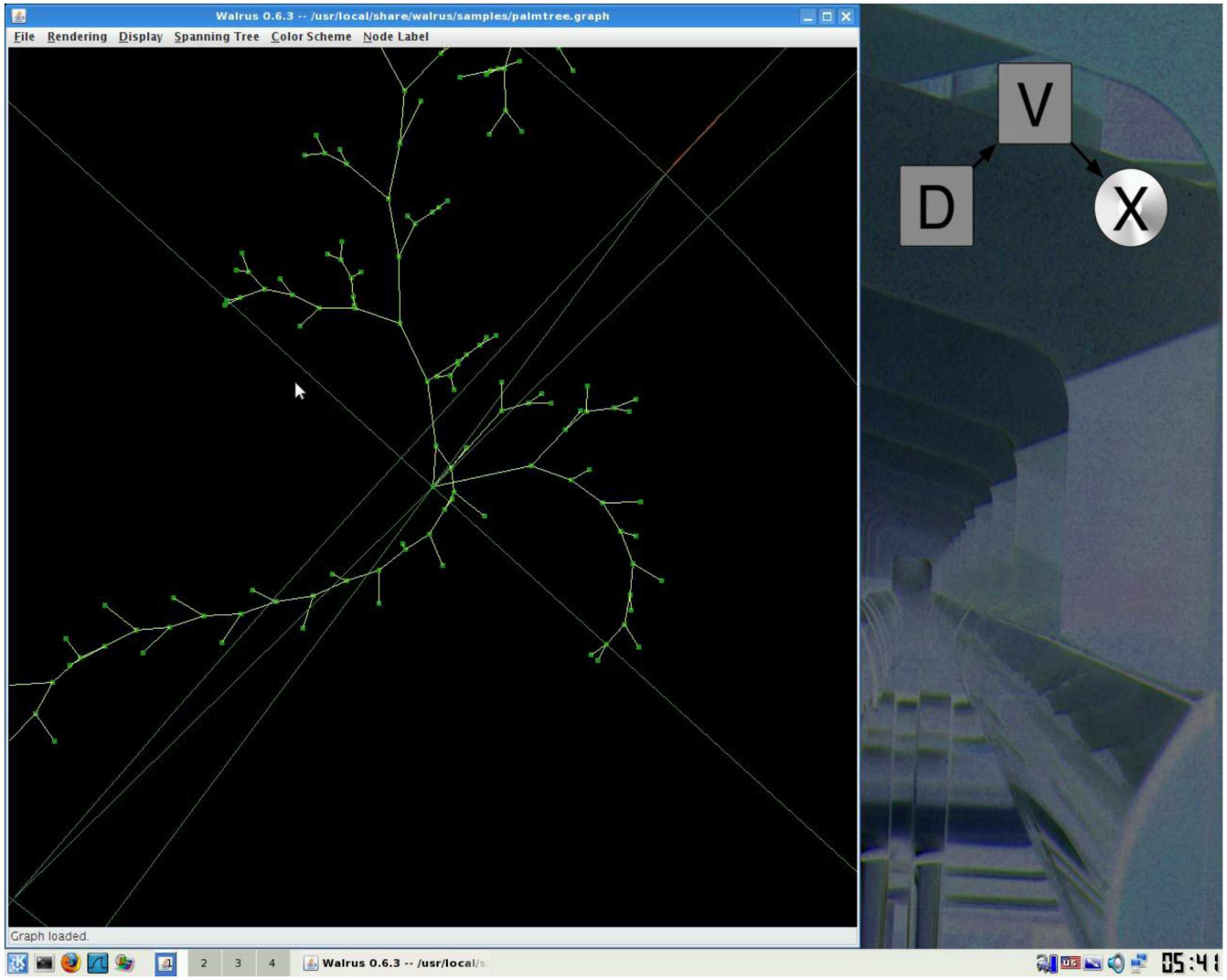


180

I went to Display -> Zoom In and selected it 4 times to get a closer view of the graph.



I used my mouse in the window and dragged the graph around to rotate it.

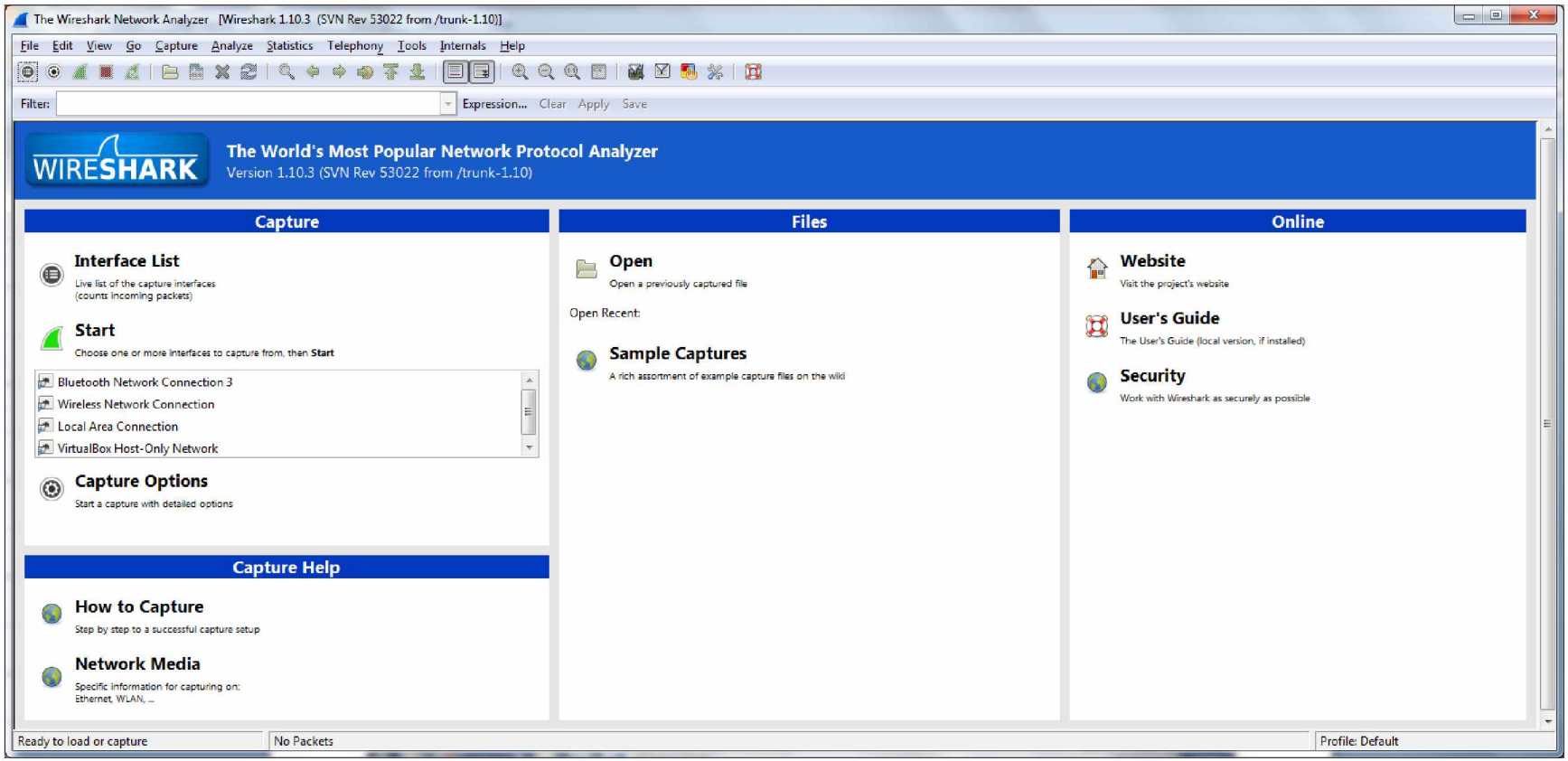


This was a very artistic representation of the data and had a number of features that made it appealing to very large datasets.

Appendix K - Tool Example – Wireshark

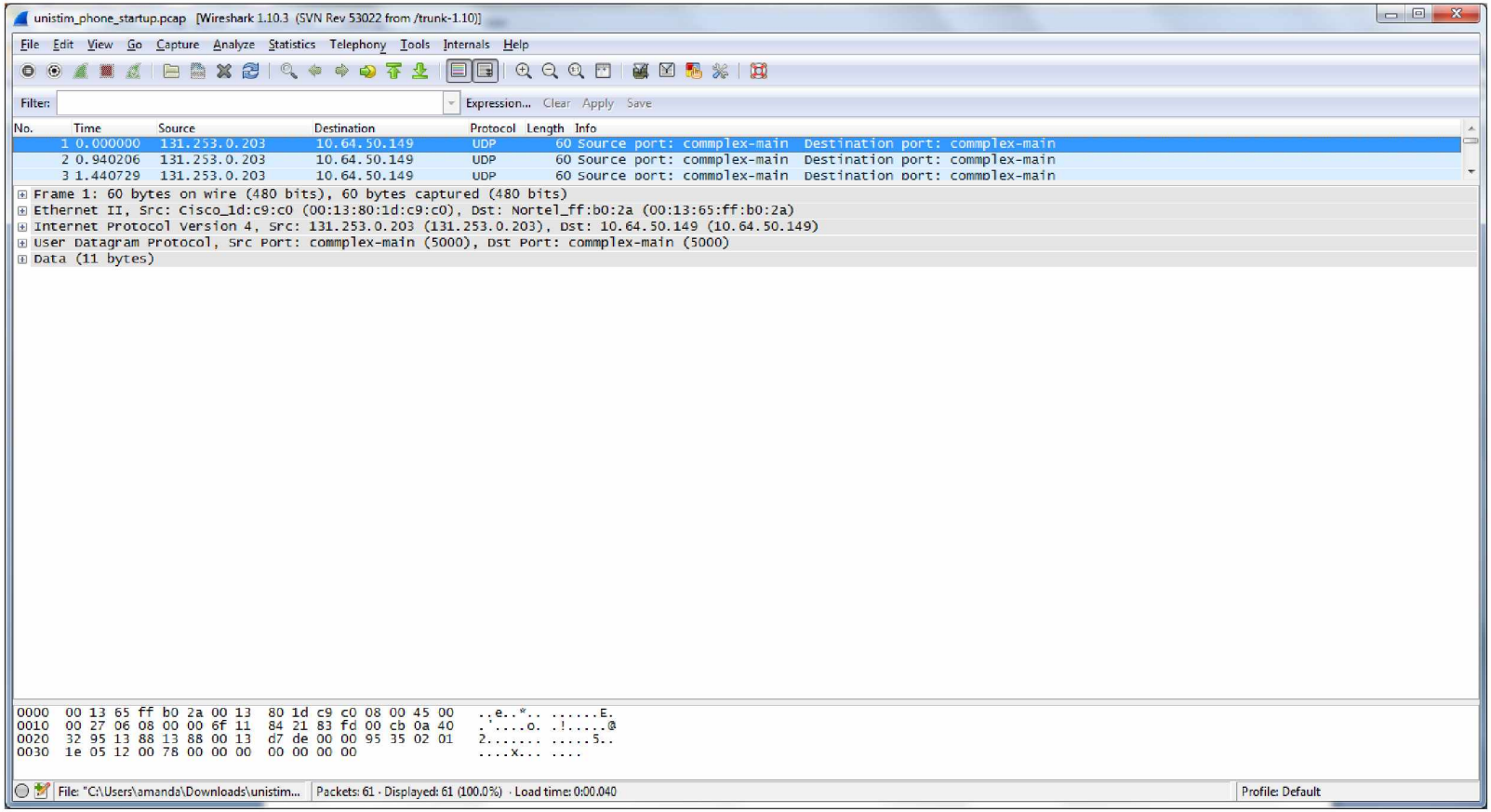
Wireshark allows capturing and viewing network traffic. It is available at http://www.wireshark.org/download.html

I ran Wireshark and was presented with the home screen:

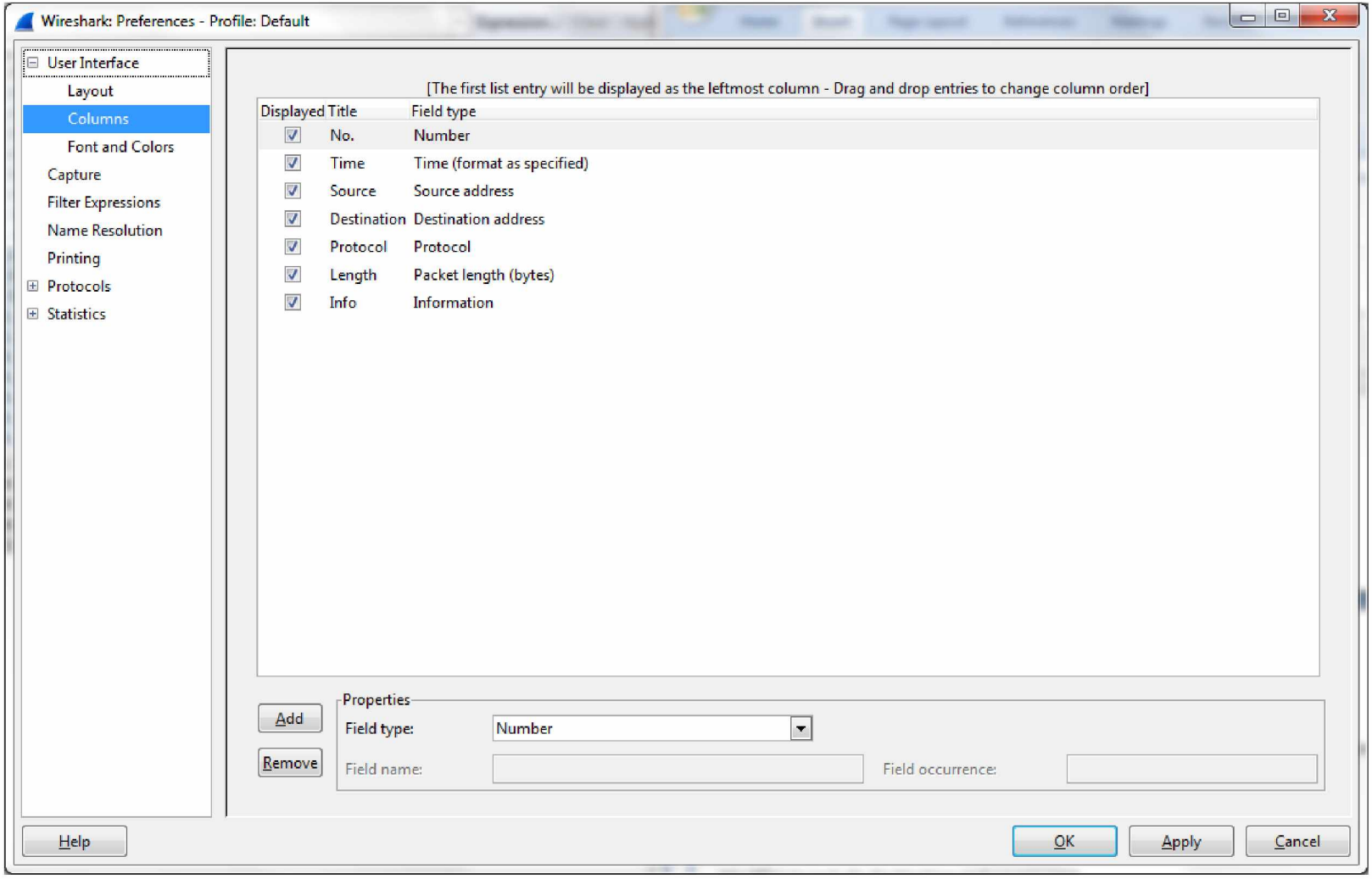


First I loaded some sample data by going to http://wiki.wireshark.org/SampleCaptures and downloaded unistim_phone_startup.pcap which is described as "Shows a phone booting up, requesting ip address and establishing connection with cs2k server."

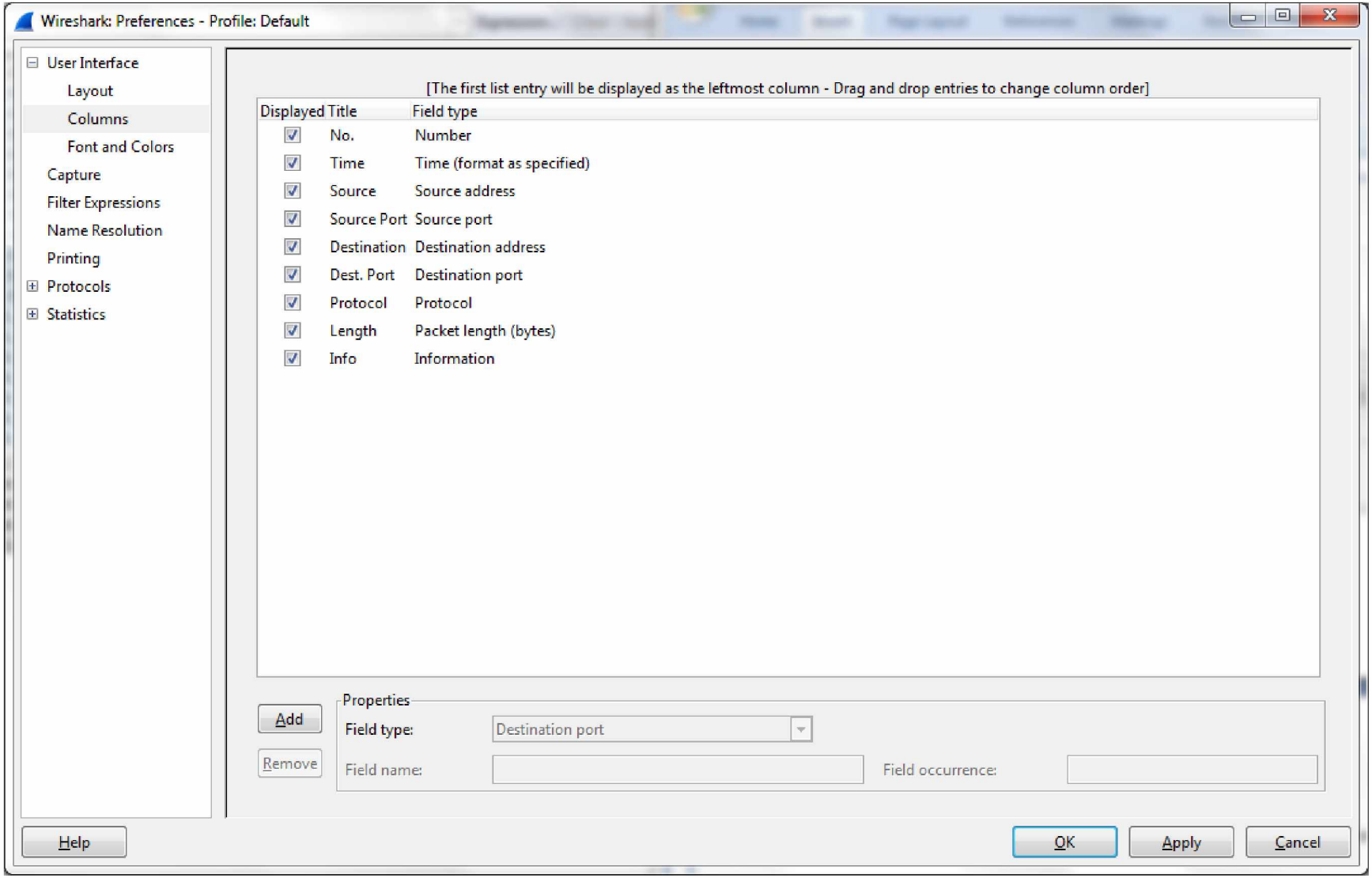
When I opened the file it immediately opened Wireshark:



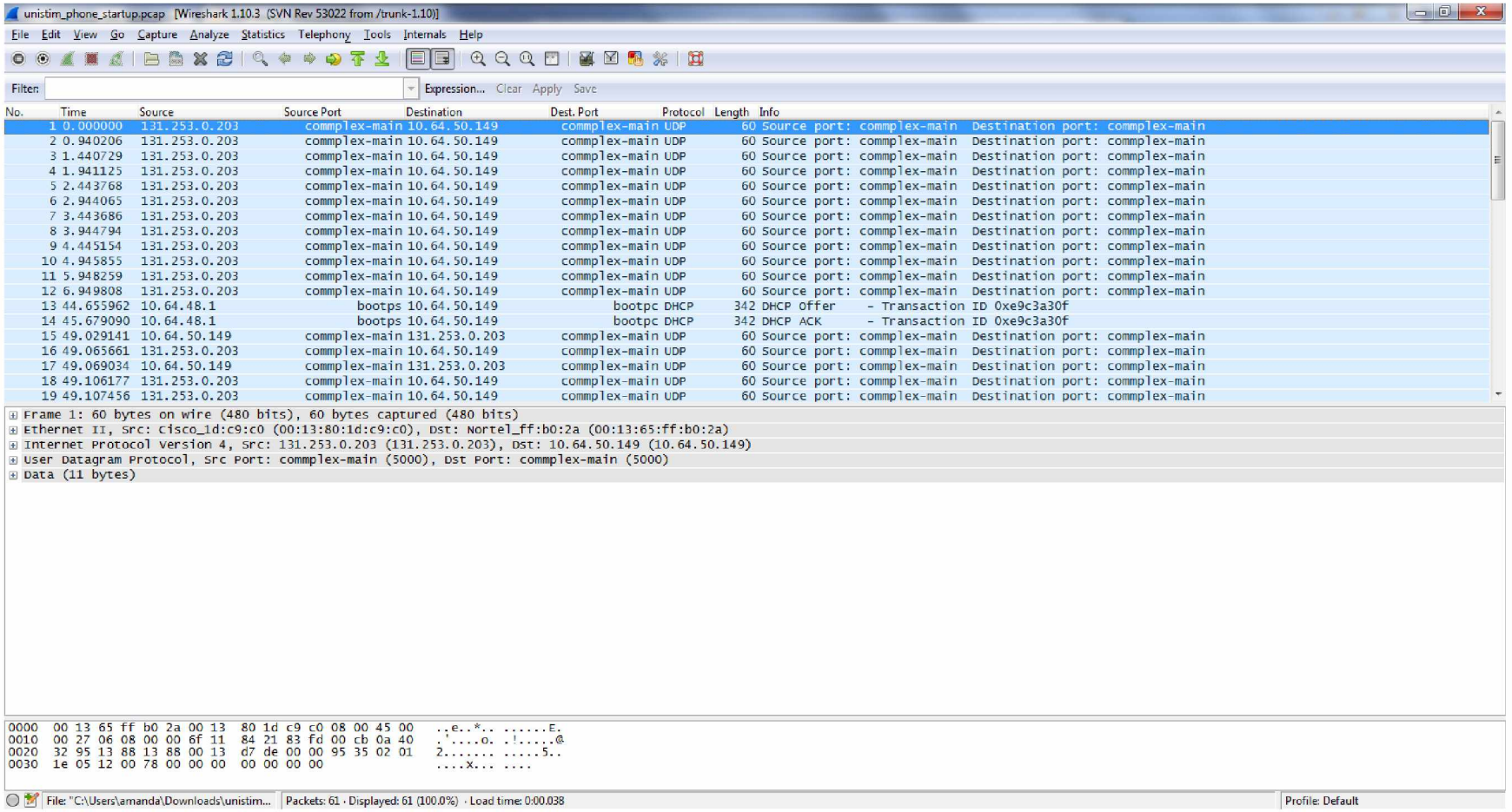
I expanded the first frame to display a longer list of the packets. I then right clicked on the menu bar and selected "Column Preferences".



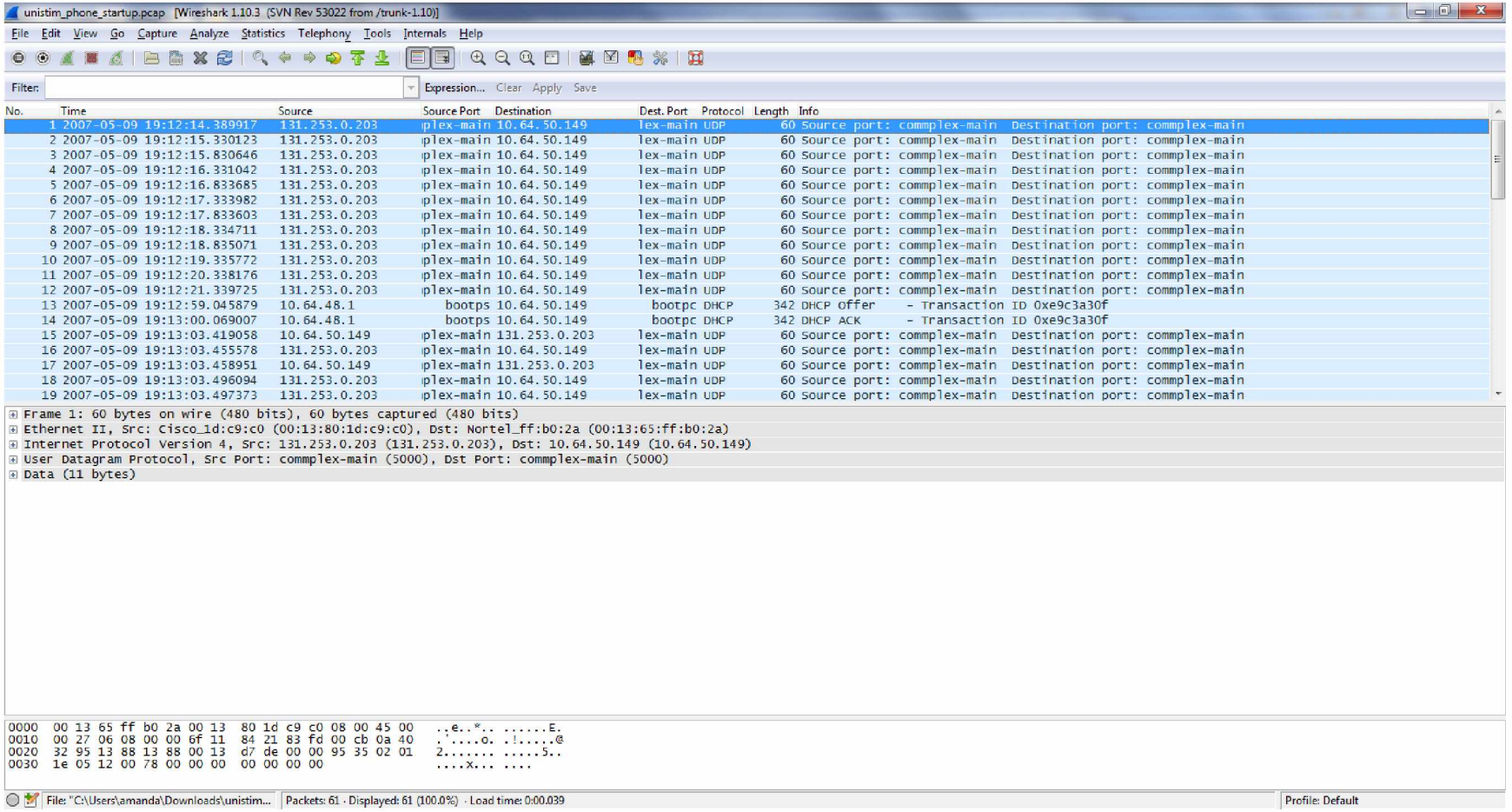
I clicked “Add” and selected Field Type: Source port. I renamed the title to “Source Port”. I did the same to add the Destination Port.



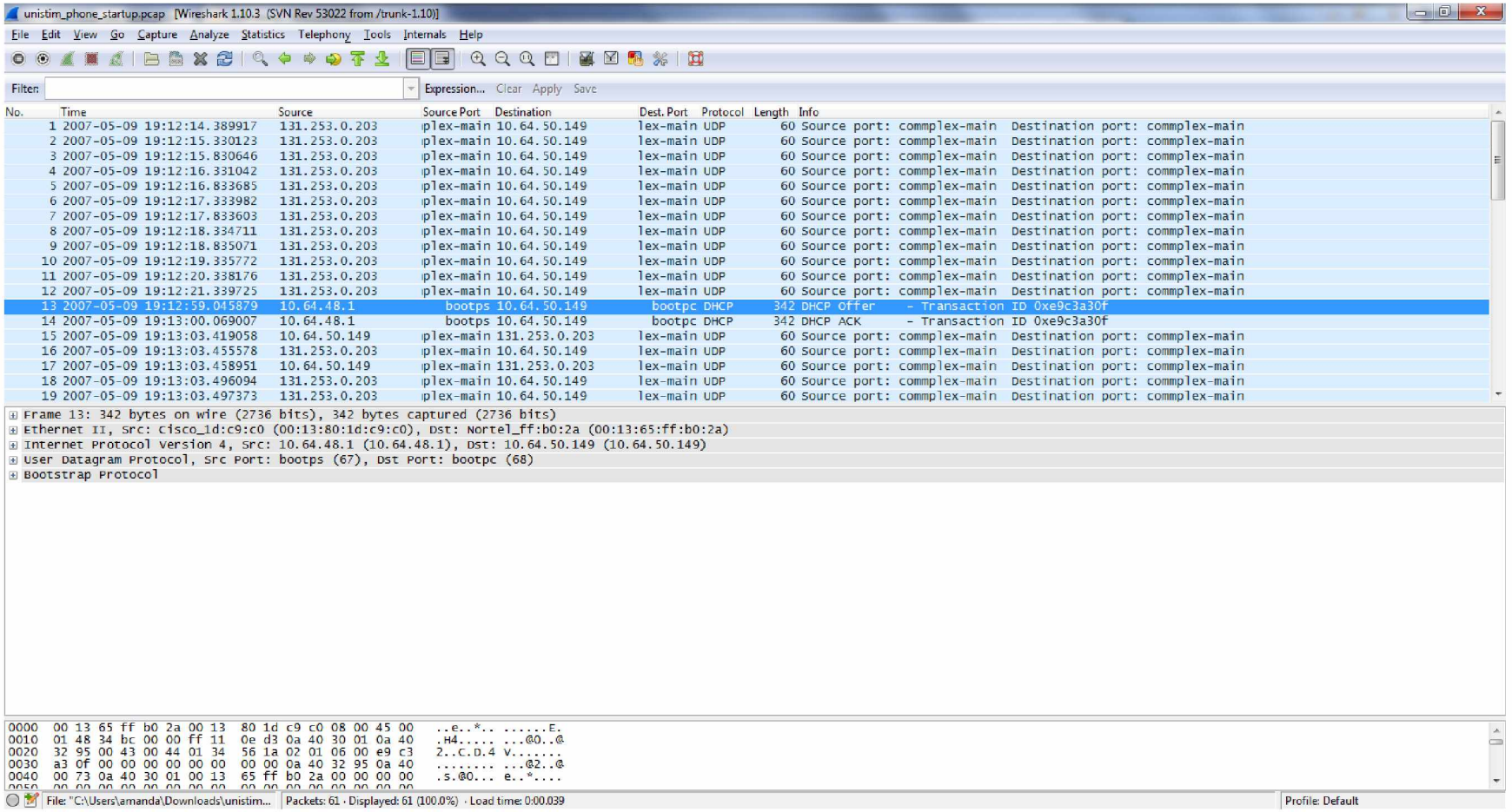
Then I clicked OK and return to the main view. The packet list is now showing more columns.



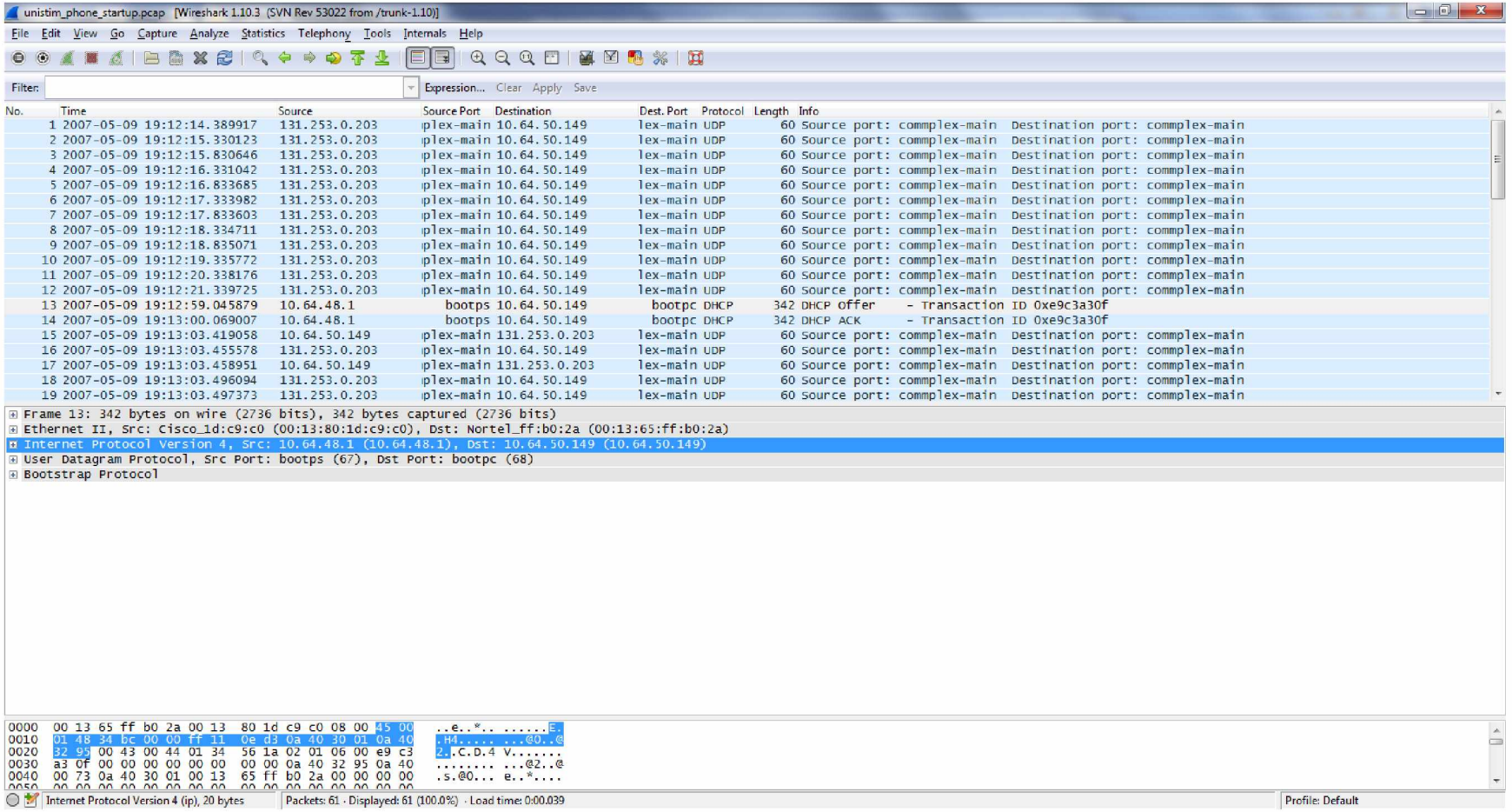
I right clicked on the menu again and selected “Column Preferences” again. I selected the “Time” column and changed the Field Type to “UTC date and time” and clicked OK.



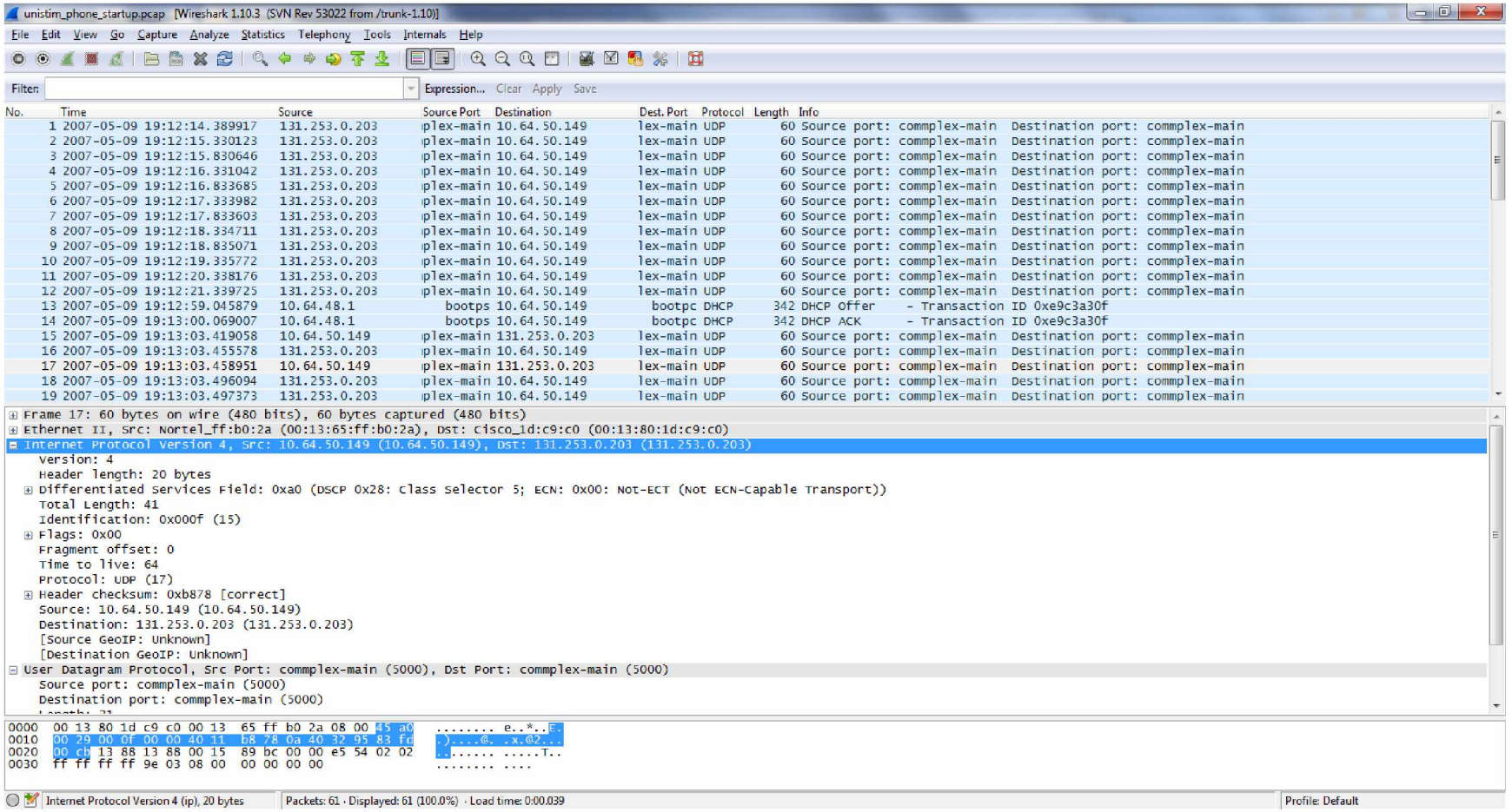
I selected packet number 13 so I could view the packet details in the frames below.



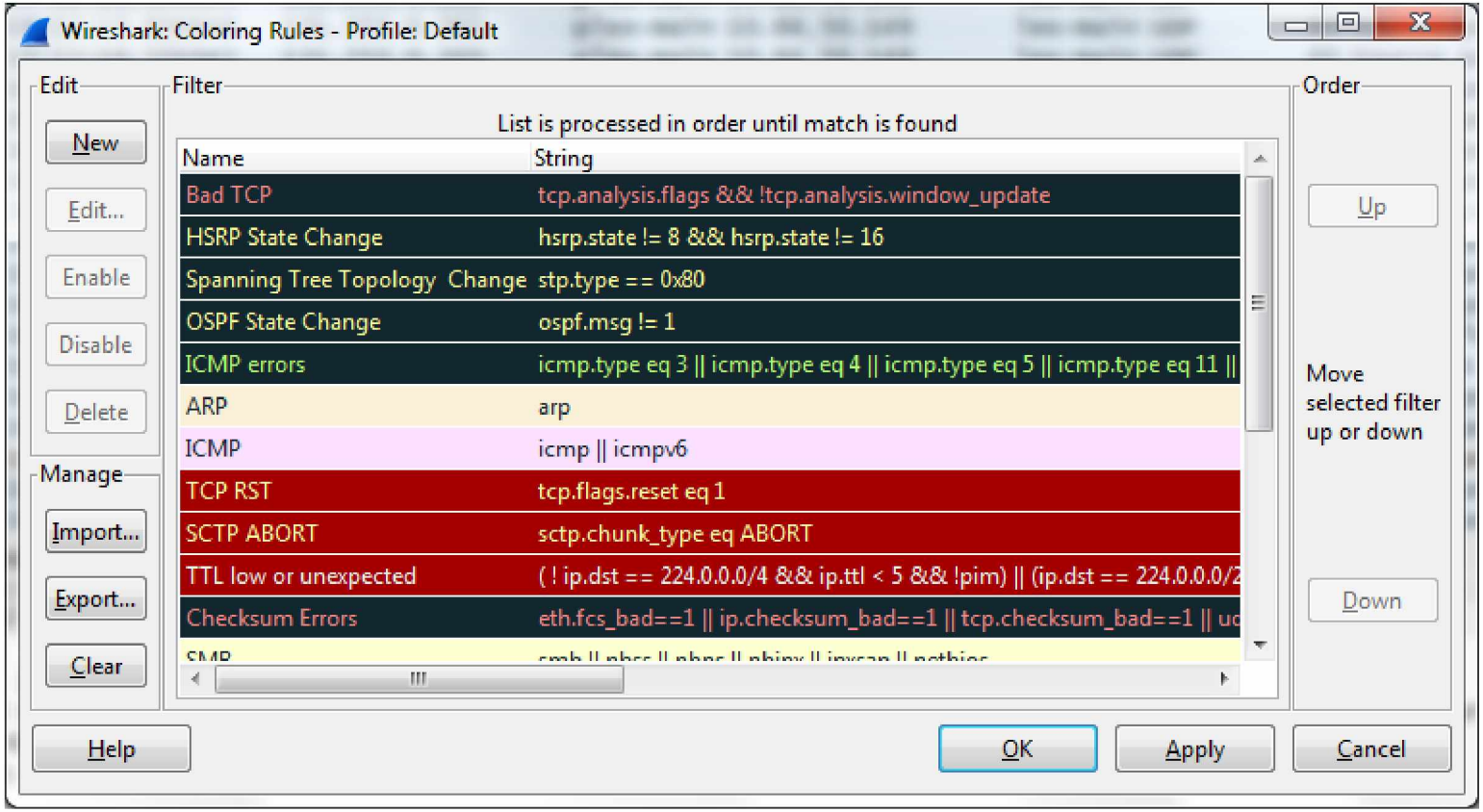
Note that when different parts of the packet details are selected then different parts of the raw data in the bottom frame are highlighted.



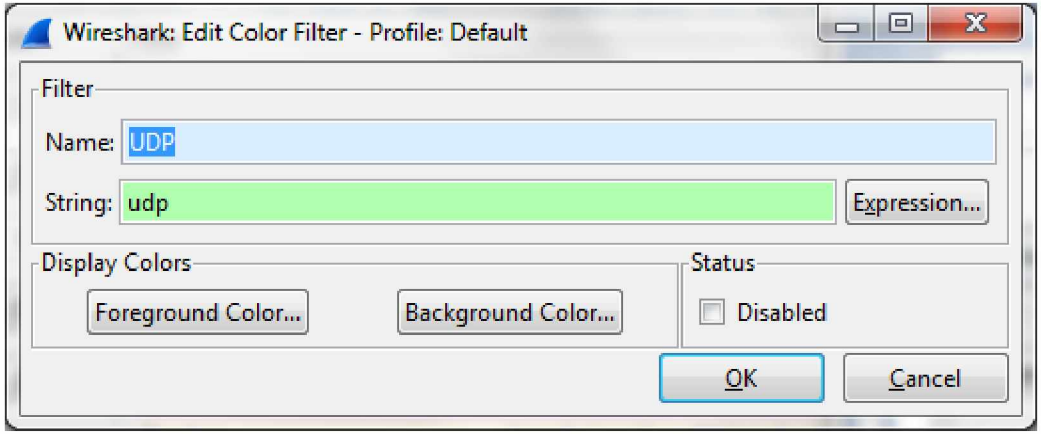
Then I selected Packet No. 17 and expanded a few of the details.



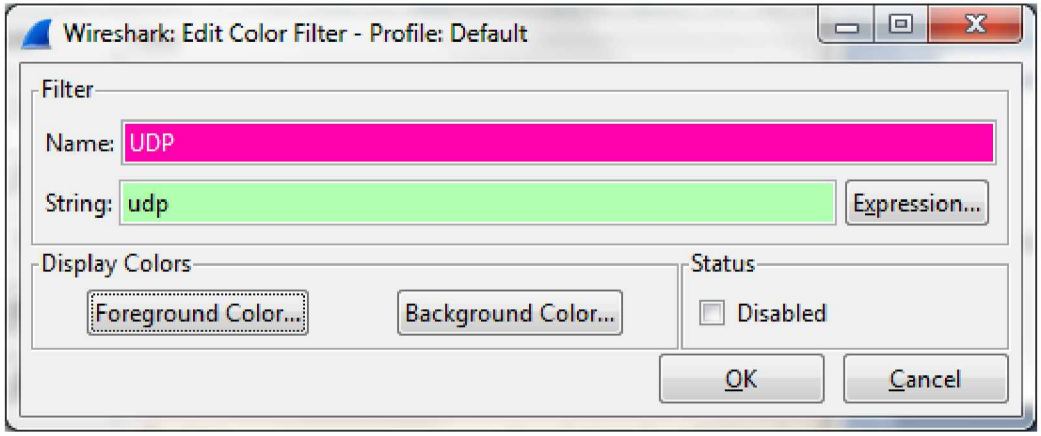
I decided to change the colors of the data types by selecting the "Edit Color Rules" icon which is the third from the left.



I scrolled down the list and selected UDP, since the sample capture has UDP packets, and clicked the Edit button on the left.

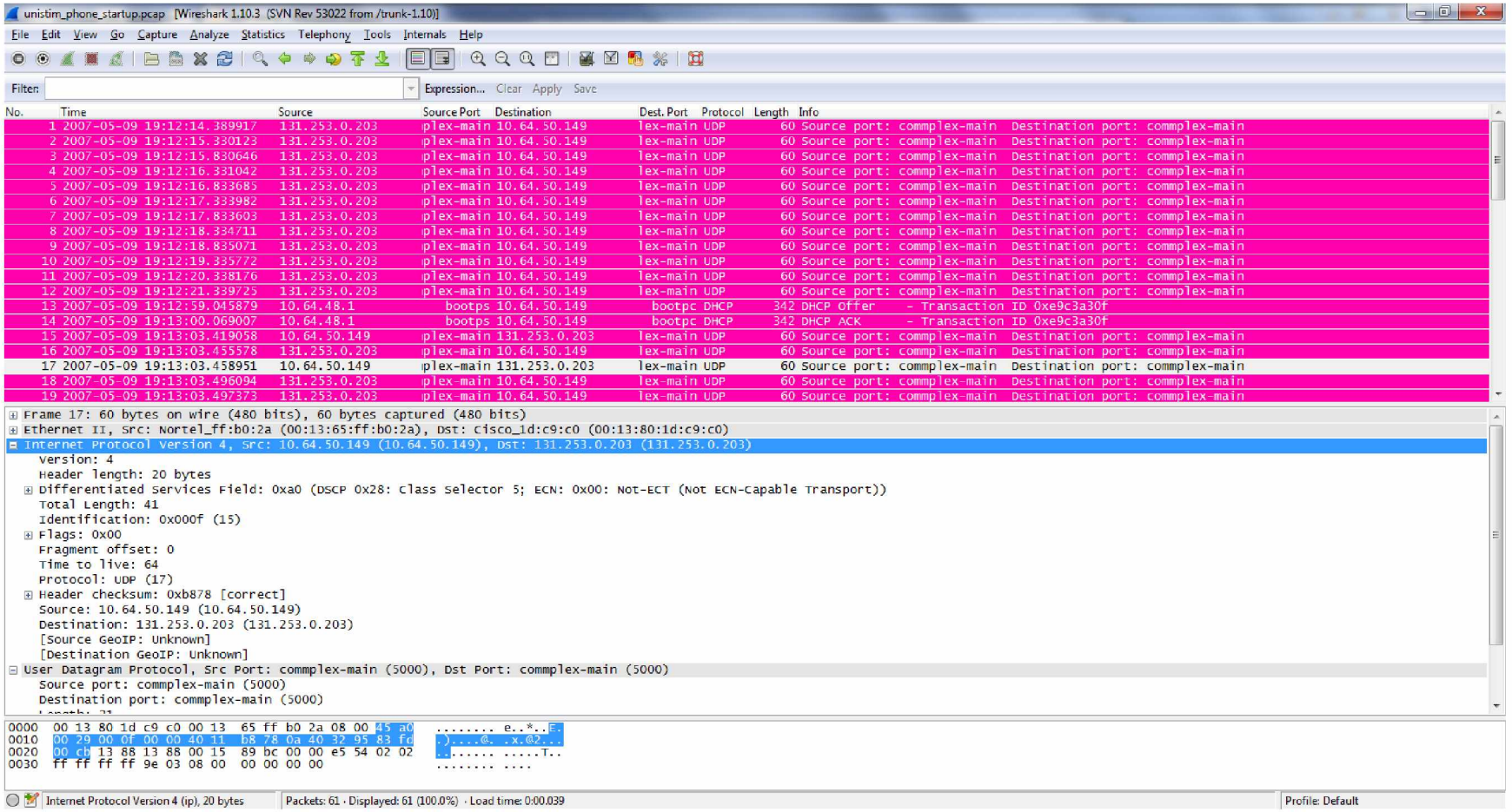


I updated the Background Color to a bright pink and the Foreground Color to white.

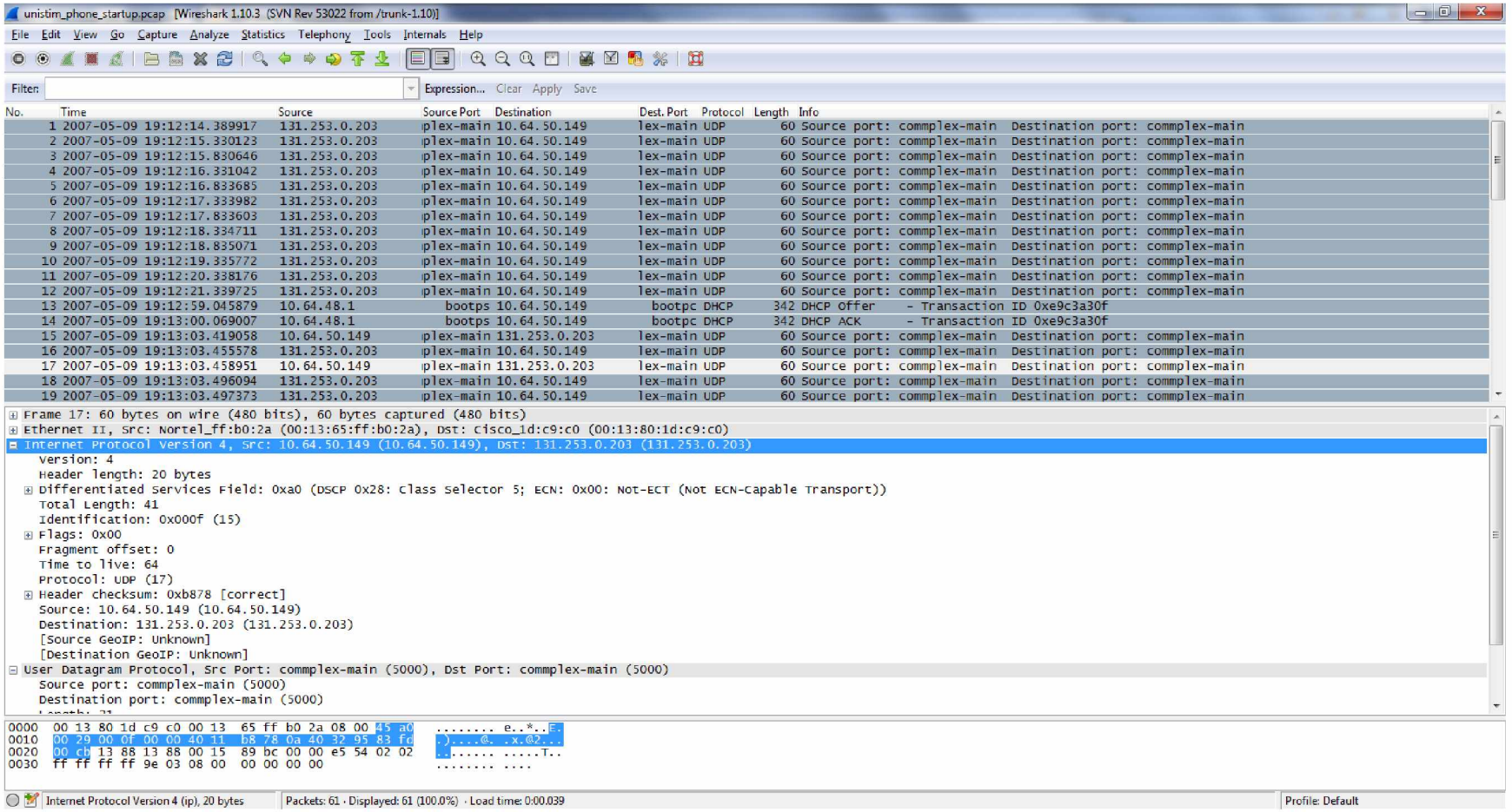


194

Then I hit OK twice. The result was a little unreadable.

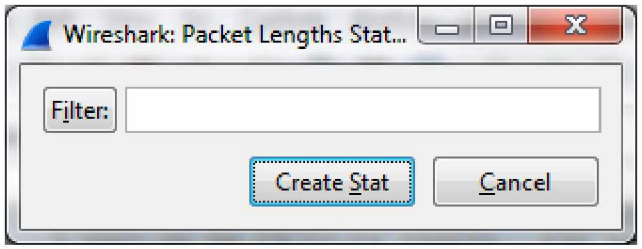


I switched the color scheme back to something manageable.

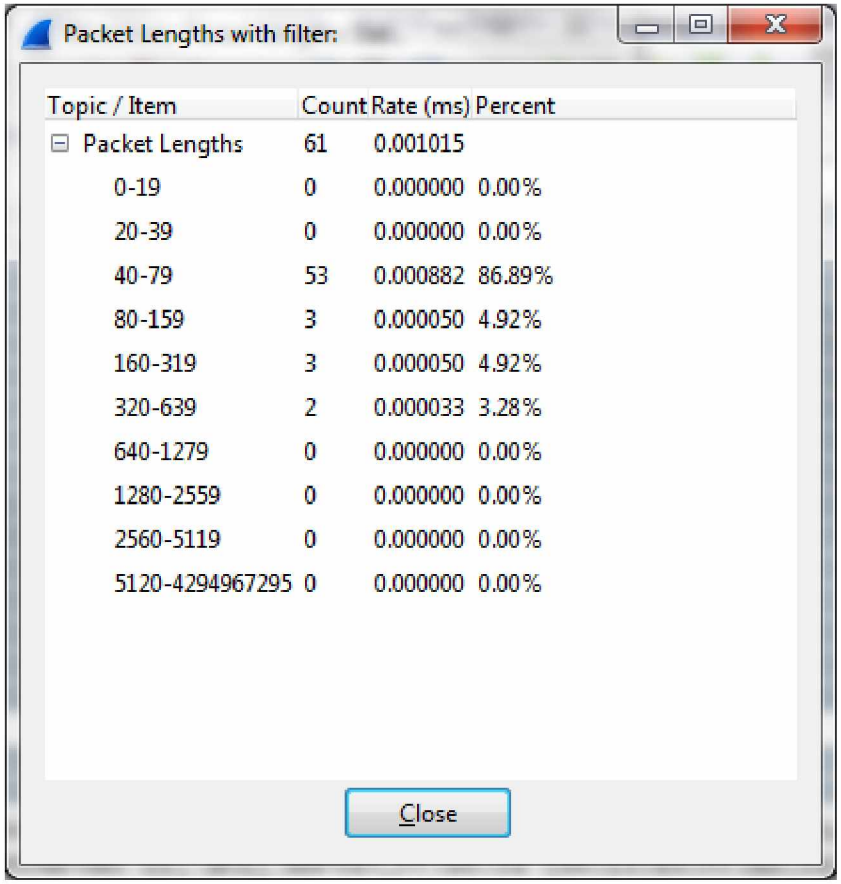


196

Then I went to the Statistics Menu and selected “Packet Lengths”. It gives you an option to filter the statistics or just “Create Stat” without a filter.

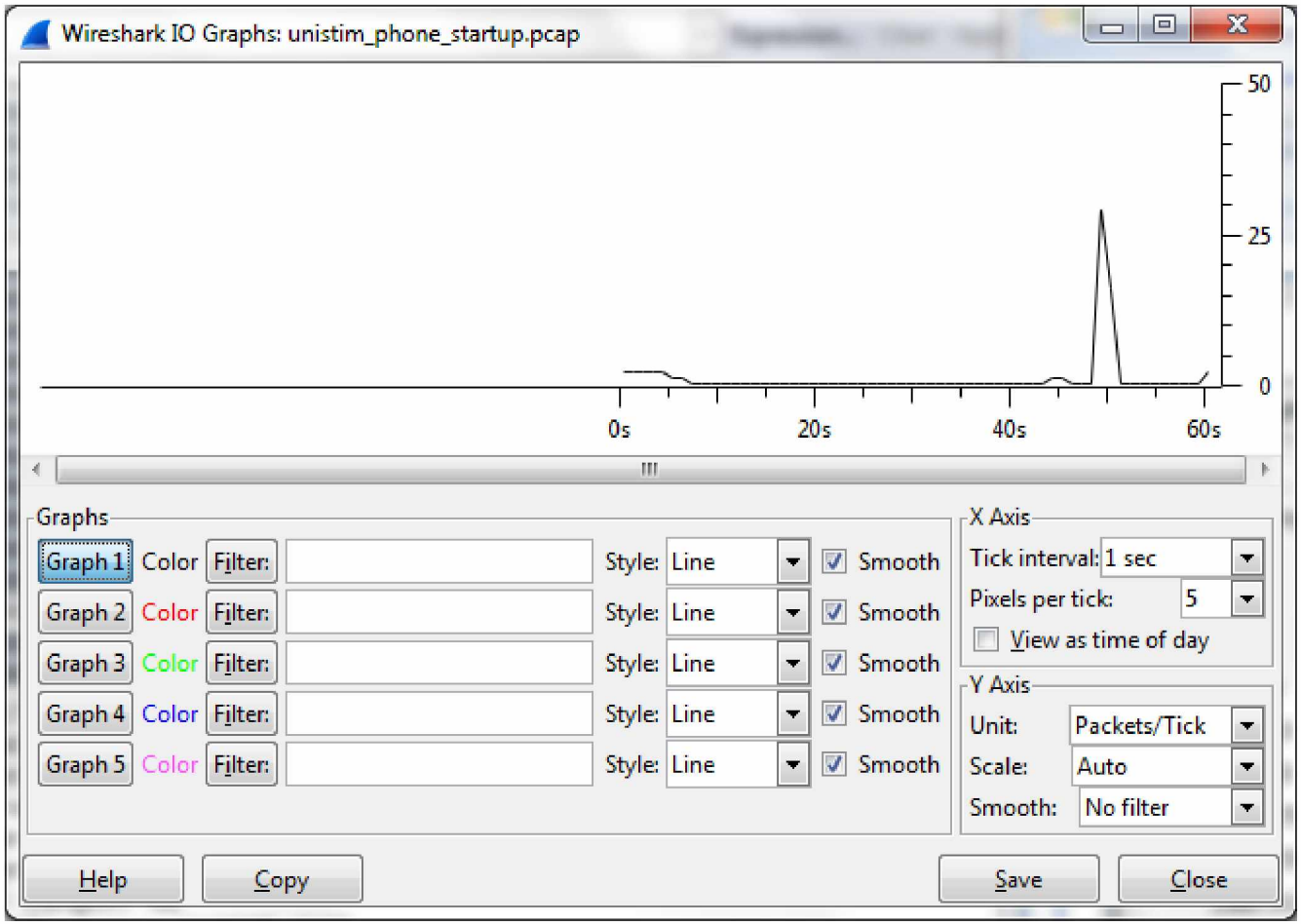


I opted for the latter and received a breakdown of the number of packets at different ranges.

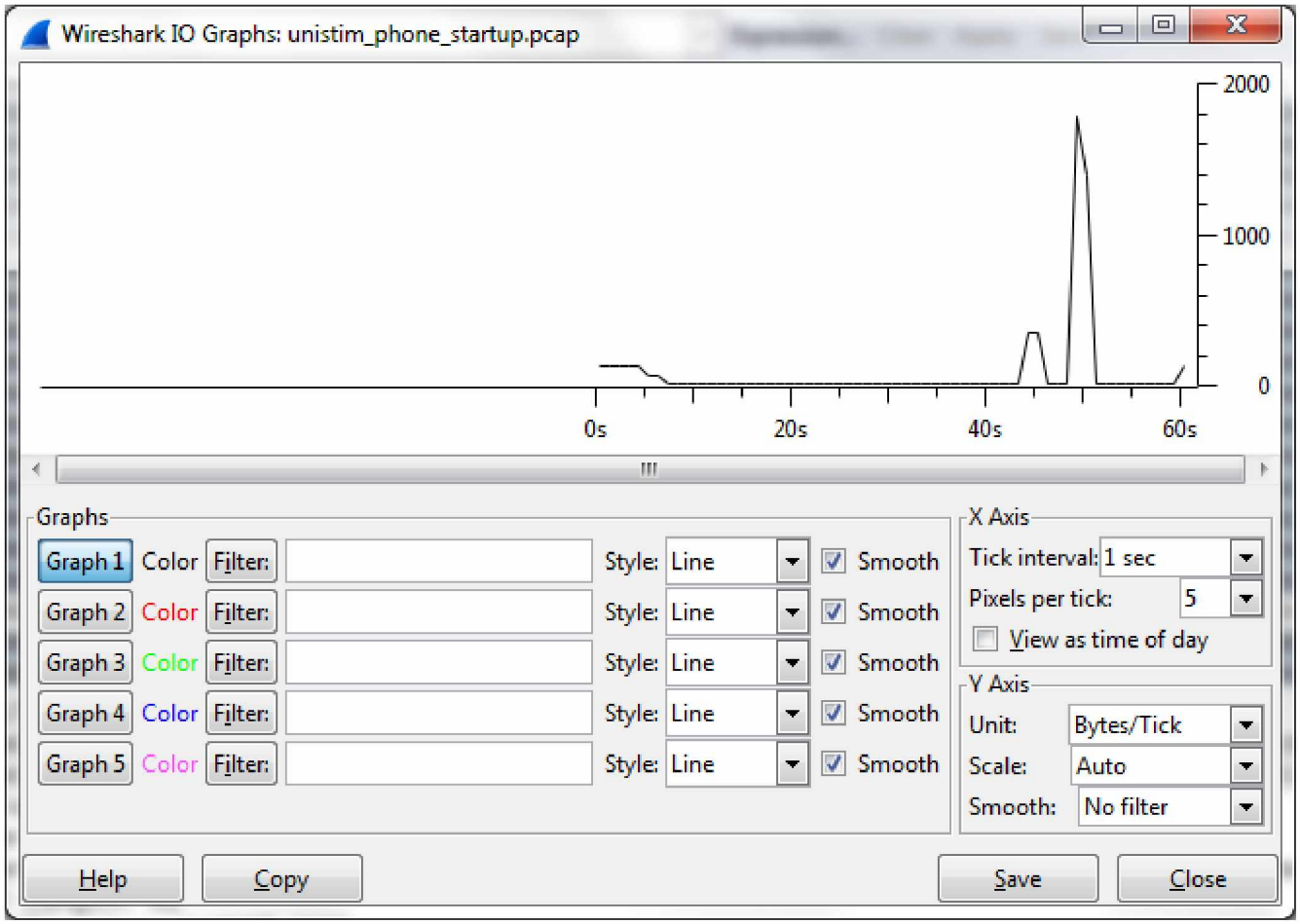


198

I then went to the Statistics Menu and selected IO Graph. The Y axis is showing the number of packets and the X axis shows a 1 second interval.

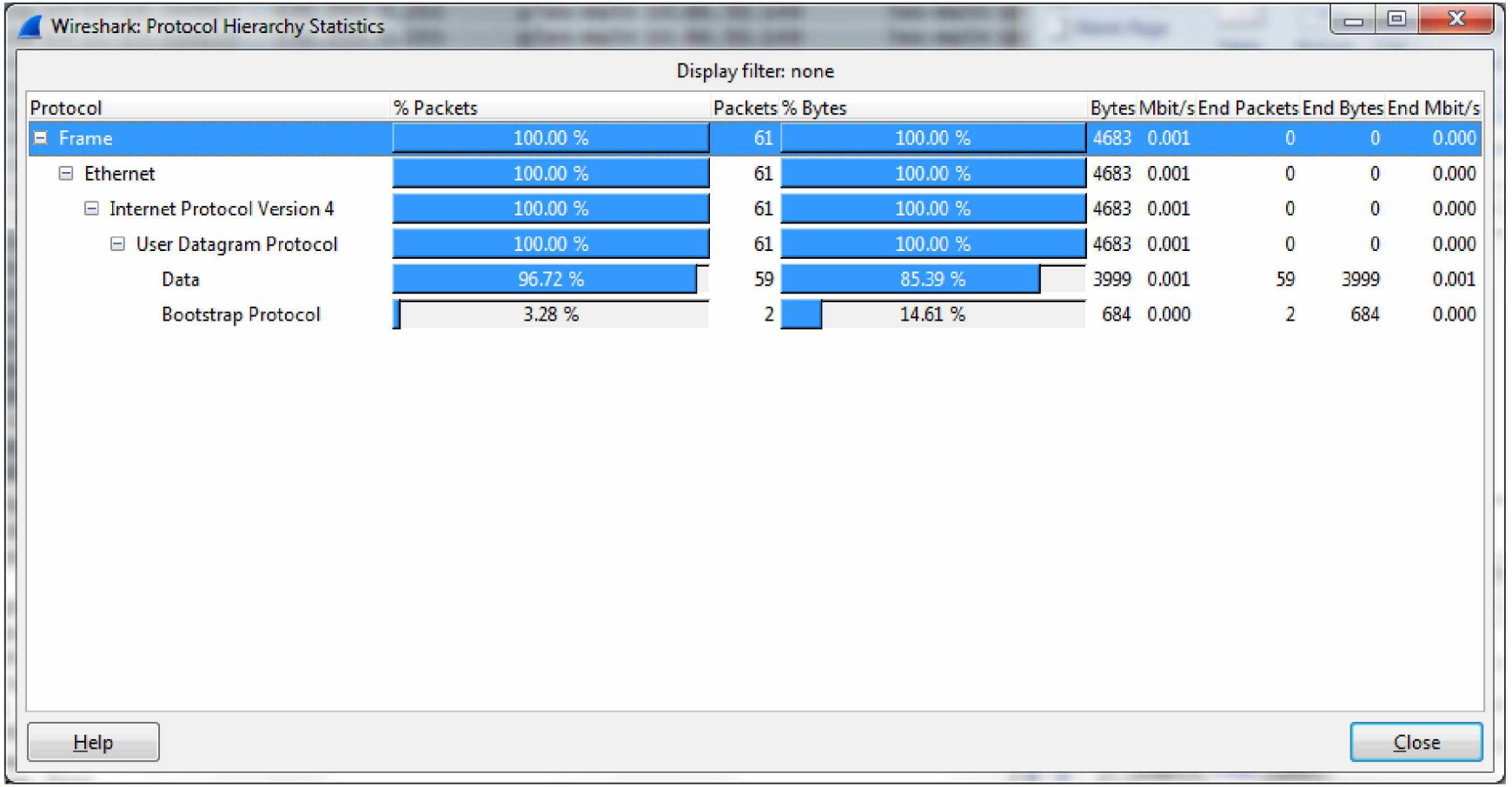


I modified the Y Axis Unit to Bytes/Tick and this resulted in a different graph.

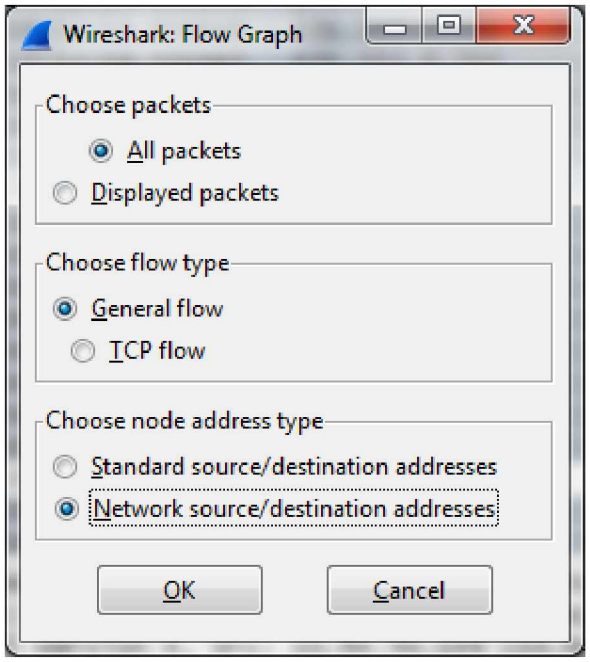


200

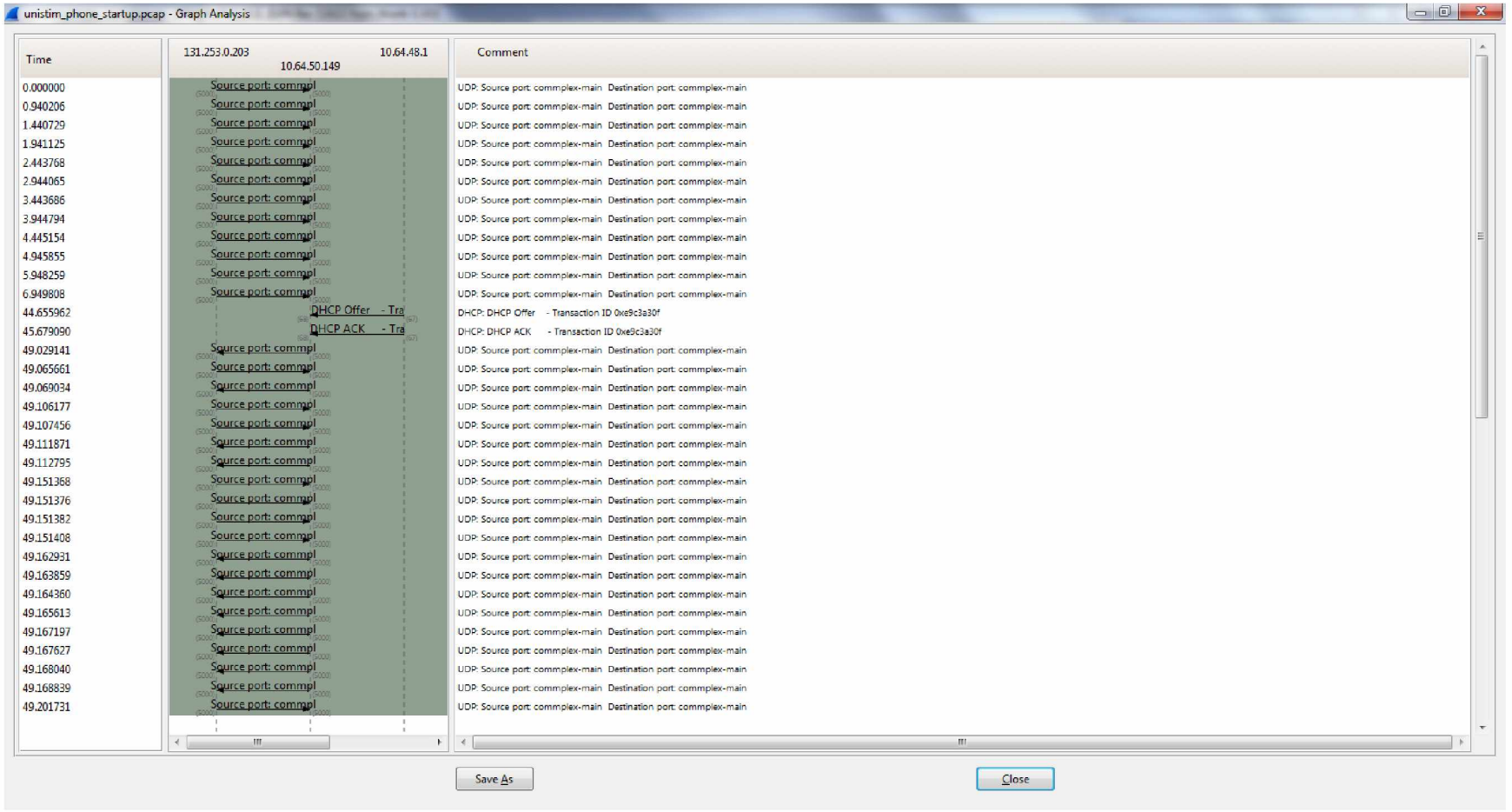
I went to the Statistics Menu and selected Protocol Hierarchy. This broke down the number of packets, bytes, and other data involved in each level of the process.



I went back to the Statistics Menu and selected Flow Graph. I changed the defaults to “All packets” and “Network source/destination addresses”.



Then I clicked OK. Since the sample capture is pretty basic there isn't much complexity to this view.



In summary, Wireshark offers an extremely wide range of tools to analyze network traffic. I barely scratched the surface in the available options. The ability to customize the tool is also excellent. It is not surprising that this is so popular.